

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) - BarcelonaTech

Implementació de xarxa social per a persones amb diversitat funcional

Treball fi de grau

Grau en Enginyeria Informàtica
Enginyeria del Software

Autor: David Aleu Moseguí

Tutor: Xavier Burgués Illa
Enginyeria de Serveis i Sistemes d'Informació

28-06-2017

Agraïments:

Agraeixo al Xavier Burgués haver acceptat ser el director del meu Treball de Final de Grau, els seus consells, ajuda i supervisió durant totes les fases del projecte, i el temps que ha dedicat per atendre'm en les reunions.

Agraeixo a Anna Maria Jornet, pedagoga i logopeda que treballa en el món de l'educació especial, i al Marc Coronas de la UTAC (Unitat de Tècniques Augmentatives de Comunicació), per haver resolt tots els meus dubtes i donar-me informació sobre el sector en què s'emmarca aquest treball.

Agraeixo als meus companys de la universitat, per fer que el transcurs del desenvolupament d'aquest projecte hagi estat més fàcil i agradable.

Finalment, agraeixo a tota la meva família el seu suport i confiança que m'han donat durant tot el període acadèmic.

Resum

La tecnologia ha estat evolucionant exponencialment des de fa uns anys. Tot hi això hi ha sectors els quals no se'ls hi ha mostrat prou d'interès. Un d'aquests sectors, és el de les aplicacions per a persones amb diversitat funcional de dèficit cognitiu.

Aquí l'objectiu d'aquest treball de final de grau, crear un *software* que no els faci sentir un públic diferent, una xarxa social adaptada per a persones que presenten aquest problema.

El treball consistirà a desenvolupar una xarxa social adaptada per a persones amb diversitat funcional de dèficit cognitiu, amb l'addició d'una eina de gestió per als professionals de l'àmbit, amb la qual podran gestionar esdeveniments de diferents organitzacions.

Resumen

La tecnología ha ido evolucionando des de hace unos años. Dados los hechos, todavía hay sectores en los que no se les muestra suficiente interés. Uno de estos sectores es el de las aplicaciones para personas con diversidad funcional de déficit cognitivo.

Aquí el objetivo de este trabajo de fin de grado, crear un *software* que no los haga sentir un público distinto, una red social adaptada para personas que presenten este problema.

El trabajo consistirá en desarrollar una red social adaptada para personas con diversidad funcional de déficit cognitivo, con la adición de una herramienta para los profesionales en el ámbito, con la que podrán gestionar eventos de distintas organizaciones.

Abstract

Technology has evolved from a few years ago. Despite this, there are still sectors where they are not shown sufficient interest. One of these sectors is the one of the applications for people with functional diversity of cognitive deficit.

Here is the purpose of this end-of-grade paper, to create a *software* that does not make them feel different, a social network adapted for people who present this problem.

The work will consist of developing a social network adapted for people with functional diversity of cognitive deficit, with the addition of a tool for professionals in the field, with which they can manage events of different organizations.

Índex

Índex	v
Índex de figures	vii
Índex de taules	vii

1 Introducció i Contextualització	1
1.1 Context	1
1.1.1 Introducció	1
1.1.2 Stakeholders	1
1.2 Estat de l'Art	2
1.2.1 Contextualització	2
1.2.2 Estudi del mercat	3
1.2.3 Conclusions de l'estudi de mercat	4
1.3 Formulació del problema	5
1.3.1 Objectiu principal	5
1.3.2 Objectius específics	5
2 Gestió del Projecte	7
2.1 Abast	7
2.1.1 Possibles obstacles	7
2.2 Metodologia i rigor	8
2.2.1 Metodologies de treball	8
2.2.2 Eines de seguiment	8
2.2.3 Metodologies de treball	8
2.3 Planificació temporal	9
2.3.1 Planificació general	9
2.3.2 Descripció de les tasques	10
2.3.3 Calendari	11
2.4 Gestió Econòmica	14
2.4.1 Consideracions inicials	14
2.4.2 Identificació i estimació de costos	14
2.4.3 Control de gestió	17
2.5 Sostenibilitat i compromís social	17
2.5.1 Econòmica	17
2.5.2 Social	18
2.5.3 Ambiental	18
3 Seguiment del Projecte	19
3.1 Introducció	19
3.2 Product Backlog	20
3.3 <i>Sprint</i> 1	20
3.3.1 Resum de l' <i>sprint</i>	20
3.3.2 Descripció i resultat de les històries d'usuari	21
3.3.3 Tecnologies utilitzades	23
3.3.4 Conclusions de l' <i>sprint</i>	25

3.4	<i>Sprint 2</i>	25
3.4.1	Resum de l' <i>sprint</i>	25
3.4.2	Descripció i resultat de les històries d'usuari	25
3.4.3	Tecnologies utilitzades	28
3.4.4	Conclusions de l' <i>sprint</i>	29
3.5	<i>Sprint 3</i>	29
3.5.1	Resum de l' <i>sprint</i>	29
3.5.2	Descripció i resultat de les històries d'usuari	29
3.5.3	Conclusions de l' <i>sprint</i>	31
3.6	<i>Sprint 4</i>	31
3.6.1	Resum de l' <i>sprint</i>	31
3.6.2	Descripció i resultat de les històries d'usuari	31
3.6.3	Conclusions de l' <i>sprint</i>	32
3.7	<i>Sprint 5</i>	32
3.7.1	Resum de l' <i>sprint</i>	32
3.7.2	Descripció i resultat de les històries d'usuari	32
3.7.3	Tecnologies utilitzades	33
3.7.4	Conclusions de l' <i>sprint</i>	34
3.8	<i>Sprint 6</i>	34
3.8.1	Resum de l' <i>sprint</i>	34
3.8.2	Descripció i resultat de les històries d'usuari	34
3.8.3	Tecnologies utilitzades	35
3.8.4	Conclusions de l' <i>sprint</i>	35
3.9	Conclusions de Seguiment	35
4	Informació Tècnica	37
4.1	Introducció	37
4.2	Disseny	37
4.2.1	Arquitectura	37
4.2.2	Tecnologia utilitzada	39
4.2.3	User Interface	41
4.2.4	Mapes navegacionals	42
4.2.5	Model de dades	43
4.3	Implementació	46
4.3.1	Sessions	46
4.3.2	Registre	48
4.3.3	New Story	51
4.3.4	Other User Profile	55
4.4	Proves	57
5	Implantació del Sistema	59
5.1	Introducció	59
5.1.1	Implantació en un servidor Linux	59
5.1.2	Lleis que apliquen	60
5.1.3	Procediment d'aprenentatge	60
6	Conclusions	61
6.1	Futurs Projectes	62
	Bibliografia	63

Índex de figures

1.1	Figura 1.1	5
2.1	Figura 2.1	13
3.1	Figura 3.1	21
3.2	Figura 3.2	22
3.3	Figura 3.3	22
3.4	Figura 3.4	27
4.1	Figura 4.1	39
4.2	Figura 4.2	41
4.3	Figura 4.3	42
4.4	Figura 4.4	42
4.5	Figura 4.5	43
4.6	Figura 4.6	43
4.7	Figura 4.7	46
4.8	Figura 4.8	48
4.9	Figura 4.9	51

Índex de taules

2.1	Taula 6.1	11
2.2	Taula 7.1	14
2.3	Taula 7.2	15
2.4	Taula 7.3	16
2.5	Taula 7.4	16
2.6	Taula 7.5	17
2.7	Taula 7.5	17
3.1	Taula 7.5	20

CAPÍTOL 1

Introducció i Contextualització

1.1 Context

1.1.1. Introducció

Aquest projecte es realitza com a Treball Final de Grau de modalitat A, dels estudis de Grau en Enginyeria Informàtica, especialitat en Enginyeria del *Software*, a la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya). El projecte en qüestió, tracta en desenvolupar una xarxa social per a persones amb diversitat funcional de dèficit cognitiu, per tal de facilitar les comunicacions que puguin tindre entre ells a través d'internet.

1.1.2. Stakeholders

Com en tots els projectes, hi haurà unes parts interessades (o stakeholders), les quals aportaran diferents objectius i visions. En aquesta secció, definirem i explicarem quines són les parts interessades i que aportaran cadascuna d'elles en el nostre projecte.

- **Director del projecte:**

El director del projecte és Xavier Burgués Illa, i serà l'encarregat de guiar i supervisar tota la feina fet per l'autor del projecte.

- **Equip desenvolupador del sistema:**

És el que s'encarregarà de tirar el projecte endavant, des del disseny de l'aplicació a la codificació. En un projecte normal, hi hauria diferents persones interpretant diferents rols (cap de projecte, analista, dissenyador, programador, etc.), però en aquest cas, l'autor del projecte serà qui exercirà cadascun d'aquests rols.

- **Usuaris:**

Els usuaris potencials del sistema són aquelles persones amb diversitat funcional de dèficit cognitiu de grau mitja/baix, els quals utilitzaran el sistema per comunicar-se entre ells de manera similar a Facebook.

Per altra banda, també hi haurà usuaris que tinguin relació en aquest àmbit, ho sigui, professionals especialitzats (logopedes, psicoterapeutes, etc.).

Pel que fa a la resta de gent, no tindrien accés a la xarxa social.

- **UTAC:**

La UTAC (Unitat de Tècniques Augmentatives de Comunicació) és un servei extern a la Facultat de Psicologia de la UB (Universitat de Barcelona), la qual ajudarà a

l'autor del projecte a entendre com s'ha de dissenyar la interfície gràfica de la xarxa social perquè els seus usuaris s'hi puguin moure més còmodament.

- **Competència:**

Indirectament, la competència també actua com a part interessada. Avui en dia existeixen moltes xarxes socials, com ara Facebook i Twitter, les quals no tenen implementades diferents capes gràfiques adaptades per a cadascun dels usuaris. Per tant, aquestes xarxes socials podrien representar una competència en el projecte actual.

1.2 Estat de l'Art ---

1.2.1. Contextualització

Una xarxa social és una estructura social composta per individus que estan lligats per un o més tipus d'interdependència. Dins del que serien les xarxes socials, podem trobar-ne de dos tipus:

- **Xarxes Socials Analògiques:**

Formades per grups de persones relacionades entre si, i que es desenvolupen sense sistemes electrònics.

- **Xarxes Socials Digitals:**

Formades per grups de persones relacionades entre si, i que es desenvolupen utilitzant sistemes electrònics.

A més, dins de les xarxes socials digitals, també tenim dos grups:

- **Xarxes Socials Digitals Horitzontals:**

Dirigides a tot tipus d'usuaris i sense cap temàtica concreta. Facebook i Twitter en serien un exemple.

- **Xarxes Socials Digitals Verticals:**

Es basen en un tema concret. Alguns exemples serien LinkedIn (xarxa social amb temàtica professional) o Pinterest (xarxa social amb temàtica d'oci fotogràfic).

Pel que fa al naixement de les xarxes socials, és una mica complex d'explicar. Des de fa molt de temps que les persones han creat comunitat per xerrar de temes que tenen en comú, però fins no fa gaires anys, no es va traslladar la idea de portar-ho a la xarxa. Des del punt de vista de la comunicació a través d'internet, podríem dir que el seu naixement és el 1971, en el moment en què s'aconsegueix enviar un correu electrònic entre dos ordinadors. El 1978, neix el BBS (Bulletin Board System) el qual permet compartir informació utilitzant un programa terminal. El 1994 hi ha el llançament de GeoCities, un servei que permet als usuaris crear les seves pròpies pàgines web i allotjar-les al núvol. El 1995, Internet arriba a 1 miló de webs diferents, i Randy Conrads crea Classmates, la primera xarxa social digital, que permet connectar antics companys d'estudi. El 1997 es llença AOL Instant Messenger, coincidint amb l'inici del blogging i el llançament de Google, que ofereix als usuaris un chat a temps real. Finalment, el 2003 es creen MySpace, LinkedIn i Facebook, que acaba essent la xarxa social amb més usuaris fins al dia d'avui.

Avui en dia hi ha moltes xarxes socials (Facebook, Twitter, Instagram, etc.), les quals ens serveixen per mantenir el contacte amb els amics que no veiem el nostre dia a dia,

compartir les coses que ens semblen interessants o facilitar la comunicació entre les persones que estan a una gran distància. La diversitat de funcionalitats que aquestes xarxes socials tenen és molt gran, tot depèn de l'ús que tu li vulguis donar.

1.2.2. Estudi del mercat

Per situar el projecte dins del marc de les solucions possibles, és necessari realitzar un estudi del mercat tenint en compte les dos necessitats que aquest té. Per una banda, s'ha d'implementar una xarxa social, cosa que implica fer un estudi de mercat de les xarxes socials digitals existents avui en dia per extreure'n les millors idees. Per l'altra banda, aquesta xarxa social s'ha d'implementar per a persones amb diversitat funcional de dèficit cognitiu de grau mitjà/baix, per tant haurem de fer un estudi de mercat de les diferents aplicacions que avui en dia tenen una interfície gràfica adaptada a la gent amb aquests problemes.

Estudi del mercat de les xarxes socials

Les xarxes socials que s'analitzen a continuació, són xarxes socials digitals, tant verticals com horitzontals, les quals no contenen cap tipus d'extensió ni adaptació per a l'ús de persones amb diversitat funcional amb diversitat funcional de dèficit cognitiu. Només hem agafat algunes de les que són més conegudes a tot el món, ja que el món de les xarxes socials és molt gran:

- **Facebook:**

Facebook és una xarxa social digital horitzontal fundada per Mark Zuckerberg. Aquesta xarxa permet afegir altres persones com amics, enviar missatges a un usuari concret, compartir enllaços d'interès, fotografies, etc. Cada usuari té un mur personal on pot publicar qualsevol tipus d'informació, i seguidament, aquesta pot ser visualitzada per la resta d'usuaris. A part d'això, també tens l'oportunitat de crear pàgines que estiguin adreçades a una temàtica concreta, que els altres usuaris poden seguir per estar actualitzats de la informació d'aquesta temàtica concreta.

- **Twitter:**

Twitter és una xarxa social que et permet enviar i llegir missatges de fins a 140 caràcters. Pots enviar missatges privats o pots fer publicacions públiques, tot i que aquestes últimes només es mostrarà als usuaris que et segueixin, ja sigui perquè les teves publicacions són d'interès o per què aquest usuari és amic teu en el món real. Normalment, els usuaris que utilitzen aquesta xarxa social, l'utilitzen per assabentar-se de les notícies més actuals o per estar a l'última moda en alguna cosa concreta.

- **Instagram:**

Instagram, a diferència de les dues xarxes anteriors, només et permet publicar fotos o vídeos i, com les altres dues, les publicacions que fas només les poden veure els usuaris que et segueixen. La gran part de la gent que utilitza aquesta aplicació són fans de la fotografia, o són persones virals en el món actual. A part, des de fa relativament poc, l'aplicació va llançar una funció més, anomenada Histories, en què els usuaris poden penjar fotos i vídeos que només són visibles durant 24 hores pels altres usuaris.

- **LinkedIn:**

LinkedIn, a diferència de les xarxes socials anteriors, aquesta és una xarxa social digital i vertical, de temàtica professional. Cada usuari de la xarxa té un mur, on es

representa tot el seu valor professional, ho sigui, les qualitats que l'usuari considera que té, les seves experiències professionals, els estudis, etc. Els altres usuaris de la xarxa poden validar les teves aptituds. Si alguna persona troba que el teu perfil és interessant, et pot enviar un missatge privat per oferir-te un treball. A més, tots els usuaris poden compartir notícies de l'àmbit professional que hagin trobat interessants.

Estudi del mercat d'aplicacions per a persones amb diversitat funcional de dèficit cognitiu

Pel que fa a les aplicacions adaptades a persones amb diversitat funcional de dèficit cognitiu, hi ha molt poca diversitat, i la gran part de les que he pogut analitzar són aplicacions per a nens i nenes amb aquesta deficiència:

- **Stimulus:**

Stimulus és una aplicació (més adient per a tablets), que conté exercicis interactius que ronden al voltant de 10 àrees funcionals. La seva idea es centra en teràpies no farmacològiques a persones que no poden accedir a aquest recurs. A més, també té l'intenció de fomentar l'ús de les TIC realitzant estimulacions cognitives. Ni que l'aplicació estigui pensada per a persones de gran edat, crec que també pot tindre un gran paper per agafar idees a l'ora de dissenyar un *software*.

- **Proyecto Azahar:**

Proyecto Azahar, és un conjunt de 10 aplicacions que ajuden a millorar l'autonomia de les persones amb autisme o altres discapacitats intel·lectuals. Utilitza pictogrames, sons i fotos per interaccionar amb la persona. A més, tant les fotos com els sons que utilitza el sistema es poden personalitzar.

- **H@z TIC:**

Programa essencialment dirigit a les persones amb Síndromes de Down i tracta de reforçar la destresa digital, tant en l'àmbit educatiu com en l'àmbit social per a una futura incorporació en el món laboral. A més, aquest programa augmenta les ganes d'aprendre de l'alumne i reforça la seva autoestima.

- **Adapro:**

Adapro és un processador de text dirigit a totes les persones que tenen algun tipus de diversitat funcional que dificulti les seves habilitats escriptura. L'usuari pot triar entre més de 10000 paraules, les quals estan representades en símbols o pictogrames.

1.2.3. Conclusions de l'estudi de mercat

Com podem veure, el món de les xarxes socials és molt gran, i les idees que es poden agafar d'aquí són moltes. En canvi, si busquem programes per a persones amb diversitat funcional de dèficit cognitiu, es fa més difícil, i la varietat que es troba és molt més reduïda.

Per tant, sí que serà fàcil poder agafar idees de les xarxes socials existents avui en dia, però pel que fa a l'adaptació a la gent amb diversitat funcional de dèficit cognitiu, serà una tasca més difícil, i segurament farà falta ajuda d'algun professional d'aquest àmbit perquè aconselli l'autor a l'hora d'implementar la interfície gràfica de la xarxa social del projecte.

1.3 Formulació del problema

Cada vegada és més la gent que pot fer ús d'internet, i això implica que cada vegada hi ha més gent que fa ús de les xarxes socials. Però no tothom té accés a aquestes o no es senten còmodes a l'hora d'utilitzar-les (actualment les xarxes socials no estan adaptades a la seva forma de pensar). A més, cal pensar en què la gent amb diversitat funcional de dèficit cognitiu que pot utilitzar aquestes xarxes, pot tindre problemes a l'hora de fer relacions socials, i això implica que han de buscar entre moltíssima gent per poder trobar algú amb qui es pugui tindre més confiança (per exemple, si algú té un problema similar al teu, sempre és molt més fàcil d'agafar-li confiança).

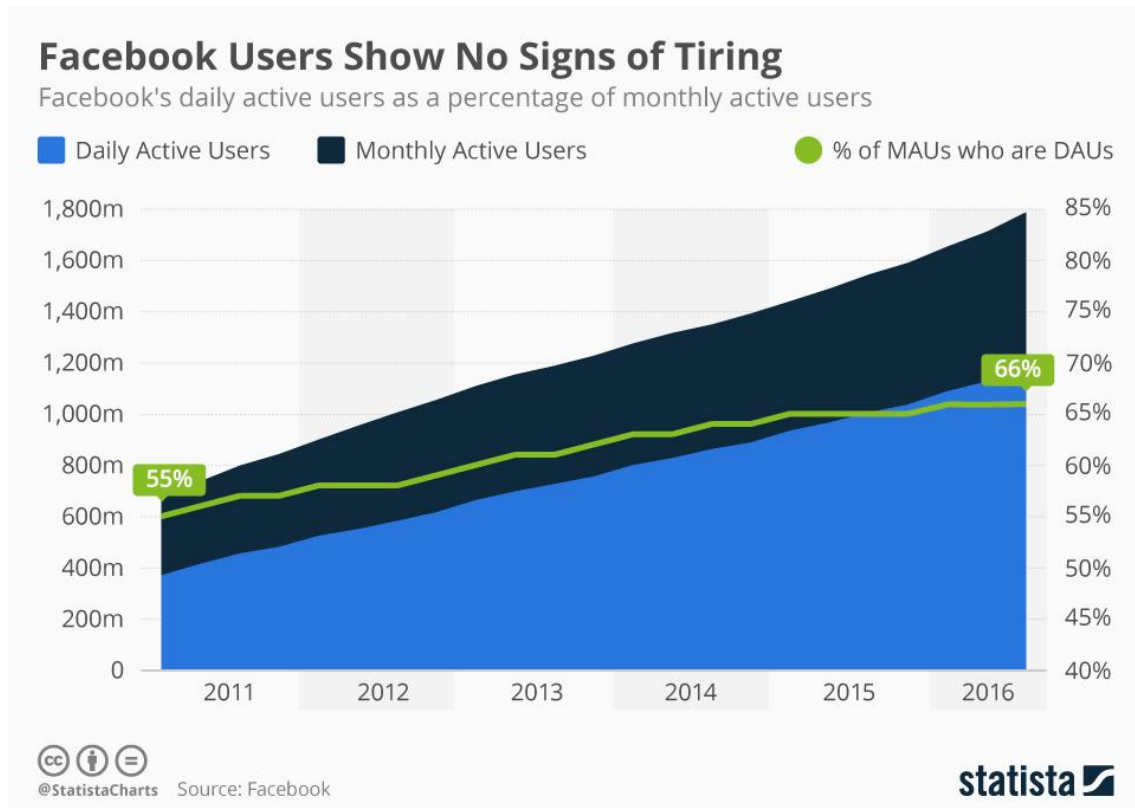


Figura 1.1: Evolució de l'activitat diària i mensual dels usuaris de Facebook.

1.3.1. Objectiu principal

L'objectiu principal del projecte és la implementació d'una xarxa social adaptada a les persones amb diversitat funcional de dèficit cognitiu de grau mitja/baix.

1.3.2. Objectius específics

A part de l'objectiu principal del projecte, tenim uns quants objectius específics, els quals alguns d'aquest, ajuden a complir amb l'objectiu principal del projecte:

- Disseny d'una interfície gràfica fàcilment manejable per persones amb baix nivell d'escriptura.
- Implementació d'una eina de gestió/monitoratge per a les associacions o professionals en l'àmbit.

- Dissenyar la xarxa social de forma que sigui *responsive* als diferents dispositius en els quals es podrà visitar.
- Implementar un sistema de seguretat, de forma que la gent que utilitzi la xarxa social siguin els usuaris amb diversitat funcional de dèficit cognitiu o professionals relacionats en l'àmbit.
- Implementar la xarxa de forma que accepti diferents idiomes.

CAPÍTOL 2

Gestió del Projecte

2.1 Abast

L'abast del projecte vindrà donat per la implementació de dos elements essencials:

- La implementació de la xarxa social per a persones amb diversitat funcional de dèficit cognitiu. Per satisfer aquest objectiu s'ha de pensar en les funcionalitats mínimes que ha de satisfer una xarxa social. Les funcionalitats mínimes que tindrà el sistema, seran les següents:
 - ◊ L'usuari podrà publicar adreces d'interès, fotografies o simplement text (possiblement acompanyat d'iconografia) al seu mur.
 - ◊ L'usuari es podrà fer amic dels altres usuaris per tal que aquests puguin veure les publicacions del seu mur.
 - ◊ L'usuari podrà marcar "M'agrada" a les publicacions dels altres usuaris (sempre i quan siguin amics).
- La implementació d'una eina de gestió per tal que els professionals de l'àmbit puguin gestionar i interactuar amb les persones que tracten en el món real. Les funcionalitats mínimes que tindrà el sistema són:
 - ◊ Possibilitat d'invitar a persones amb diversitat funcional de dèficit cognitiu a entrar al sistema.
 - ◊ Possibilitat de veure les publicacions de tots els usuaris que ha invitat.

2.1.1. Possibles obstacles

Com en qualsevol projecte hi poden haver diversos obstacles, els quals poden dificultar el desenvolupament del *software*. Els obstacles possibles serien els següents:

- **Temps limitat:**

Aquest projecte té un temps bastant limitat, ja que té una data d'entrega fixa. Si no es té una bona organització del que s'ha de fer en cada moment, poden aparèixer imprevistos que puguin paralitzar el projecte. Per això, és necessari tindre una bona planificació, deixant un marge de temps per possibles imprevistos que puguin aparèixer.

- **Desconeixement de les tecnologies utilitzades:**

Per desenvolupar el sistema s'utilitzaran tecnologies web i bases de dades no relacionals (Graph Database). Si aquestes no es coneixen prou, pot comportar que el desenvolupament del sistema sigui més lent i sense la qualitat suficient.

- **Accés a internet:**

L'usuari del sistema necessitarà internet per poder fer-ne ús. Si aquest no té internet, aleshores no tindrà accés a la xarxa social i per tant no podrà interaccionar amb ningú mitjançant aquest sistema.

2.2 Metodologia i rigor

2.2.1. Metodologies de treball

Per desenvolupar el projecte, s'utilitzarà scrum, el qual tractarà de definir diferents històries d'usuari (les quals representaran funcionalitats del sistema o necessitats del desenvolupador) amb uns certs punts d'història. Seguidament, es fixaran diferents dates en les quals, s'haurà d'haver acabat les històries d'usuari definides anteriorment (sempre per ordre de prioritat que el desenvolupador trobi oportú).

Sí que és veritat que les metodologies àgils van molt encarades als treballs en grup, però considero que tot i ser un treball individual es poden aprofitar moltes idees per tal que el projecte aconsegueixi els seus objectius. Per exemple, el desenvolupament de cada funcionalitat amb completa independència de les altres.

Les metodologies clàssiques (normalment en cascada), queden descartades, ja que el sistema que s'ha de desenvolupar tindrà moltes funcionalitats individuals, i la tecnologia en cascada, moltes vegades, implica fer canvis a les altres funcionalitats per tal d'aconseguir-ne una de nova.

2.2.2. Eines de seguiment

En tractar-se d'un projecte *software*, considero que les eines de control de versions (Github i Bitbucked) són essencials. D'aquesta forma garantim que sempre tenim el codi disponible i que podem recuperar el codi anterior en cas que tinguem errors en les noves implementacions.

2.2.3. Metodologies de treball

Per validar l'estat del projecte es passaran Unit Test, el qual tracta en testejar cadascuna de les funcionalitats del sistema un cop desenvolupades.

A part, hi haurà varies reunions amb el director per tal de garantir que el projecte segueix el camí establert inicialment.

2.3 Planificació temporal

2.3.1. Planificació general

Durada estimada del projecte

La durada estimada del projecte és de 6 mesos i mig, tenint en compte que el seu principi es troba en el moment en què vaig començar a fer recerca sobre el tema a tractar, principis de Desembre del 2016, i definint com a data final el dia de la defensa, que és a finals de Juny del 2017.

Recursos

Els recursos previstos per la realització d'aquest projecte són:

- Recursos personals: Una persona, amb una dedicació al projecte de 25 hores setmanals.
- Recursos materials: Els recursos que s'utilitzaran en aquest projecte són els següents:
 - ◇ **Control de versions:** Github o Bitbucked.
 - ◇ **Servidor:** Perquè la pàgina estigui oberta les 24 hores del dia i els seus usuaris en puguin fer ús. A més, és on es trobarà la base de dades.
 - ◇ **Correu Electrònic:** Per poder comunicar-me amb la gent que em faci falta per poder avançar el projecte.
 - ◇ **WebStorm:** *Framework* de JetBrains per al desenvolupament Web.
 - ◇ **Neo4j:** Base de dades no relacional (Graph Database), on es guardarà tota la informació dels usuaris.
 - ◇ **GoogleDocs:** Eina d'edició de text que ens servirà per fer tota la documentació del projecte.

Pla d'acció i valoració d'alternatives

La planificació del projecte es preveu flexible i modificable segons el ritme de desenvolupament del projecte.

Es poden produir desviacions temporals en els diferents *sprints* o en el temps de dedicació a aprendre el funcionament d'una eina nova. Com que utilitzem una metodologia àgil, la qual implica crear *sprints* amb històries d'usuari (representen funcionalitats individuals del sistema i es mesuren en punts d'història), si una història no s'ha pogut acabar per un *sprint*, es ficarà pel següent. En el cas que els punts d'història que no s'han complert en un *sprint* siguin molt elevats, s'haurà de fer una replanificació del projecte, retallant alguna de les funcionalitats establertes a l'inici, per tal de tenir temps d'acabar-lo.

Tot hi això, també es planificarà un *sprint* opcional, que ens servirà per desenvolupar funcionalitats extra en cas que les dates del projecte s'estiguin complint.

Consideracions globals

Encara que s'utilitzi una metodologia àgil, el projecte serà dut a terme per una sola persona, cosa que implica que no hi haurà desenvolupament de tasques en paral·lel. A més,

es farà una llista de tasques, les quals és duran a terme seqüencialment, ordenades per prioritat del desenvolupador.

Finalment, com que només hi haurà una persona que gestioni tot el projecte, el camí crític per satisfer totes les tasques és l'únic camí, i per tant, trobem innecessari fer un diagrama de Pert.

2.3.2. Descripció de les tasques

Posada en marxa

En aquesta fase, l'autor farà recerca de la informació sobre el domini del projecte, necessària per al desenvolupament, i es prepararà l'entorn de treball. A més, es parlarà amb les parts interessades necessàries per resoldre els dubtes que es puguin ocasionar. Aquesta fase no té cap dependència de precedència.

Gestió del Projecte (GEP)

Aquesta fase consta de l'elaboració del document de gestió del projecte a partir de l'assignatura GEP. Aquesta fase no té cap dependència de precedència, però sí que té un calendari estricte a seguir, ja que totes les entregues vénen donades en unes dates fixes.

Desenvolupament de la xarxa social

Elaboració de la primera part del producte, amb totes les fases del cicle de vida del desenvolupament d'un *software* incloses. Aquesta tasca consistirà a desenvolupar tota la xarxa social, en la qual ja podrà ser usable pels usuaris. A més, es seguirà el següent seguit de fases: Planificació, Disseny, Implementació i Testeig. Les dependències de precedència d'aquesta etapa són les dues anteriors (Posada en marxa i Gestió del projecte).

Desenvolupament de l'eina de gestió

Elaboració de la segona part del producte, amb totes les fases del cicle de vida del desenvolupament d'un *software* incloses. Aquesta tasca consistirà a desenvolupar tota l'eina de gestió de la xarxa social, en la qual ja podrà ser usable pels professionals en l'àmbit. A més, es seguirà el següent seguit de fases: Planificació, Disseny, Implementació i Testeig. La dependència de precedència d'aquesta etapa és l'anterior (Desenvolupament de la xarxa social).

Millores

Opcional. Última tasca de desenvolupament del producte, on es contemplarà millores de funcionament o l'addició funcionalitats secundàries. Les dependències de precedència d'aquesta etapa són les dos anteriors (Desenvolupament de la xarxa social i Desenvolupament de l'eina de gestió).

Documentació i Defensa

Aquesta fase consistirà a acabar tota la documentació (memòria del projecte) i revisar que aquesta sigui correcta. La documentació s'anirà fent en les fases anteriors. A part, també

es prepararà la defensa del projecte.

La memòria del projecte inclourà tota la documentació redactada en el mòdul de GEP i la documentació derivada de les fases anteriors. El document final i la defensa tenen com a precedència totes les tasques anteriors del projecte.

2.3.3. Calendari

Estimació d'hores

Tasca	Hores
Posada en Marxa	20
Gestió del Projecte (GEP)	70
Definició de l'abast i contextualització	24,5
Planificació Temporal	8,25
Gestió Econòmica i sostenibilitat	6,25
Presentació preliminar	6,25
Plec de condicions (Bloc Especialitats)	6,5
Document final i presentació oral	18,25
Desenvolupament de la xarxa social	300
Planificació	100
Disseny	50
Implementació i Testing	150
Desenvolupament de l'eina de gestió	100
Planificació	20
Disseny	20
Implementació i Testing	60
Millores	60
Planificació	10
Disseny	10
Implementació i Testing	40
Documentació i Defensa	60
Redacció de la Memòria	40
Preparació Defensa	20
Total	610

Taula 2.1: Estimació d'hores

Diagrama de Gantt

[illegible][illegible]

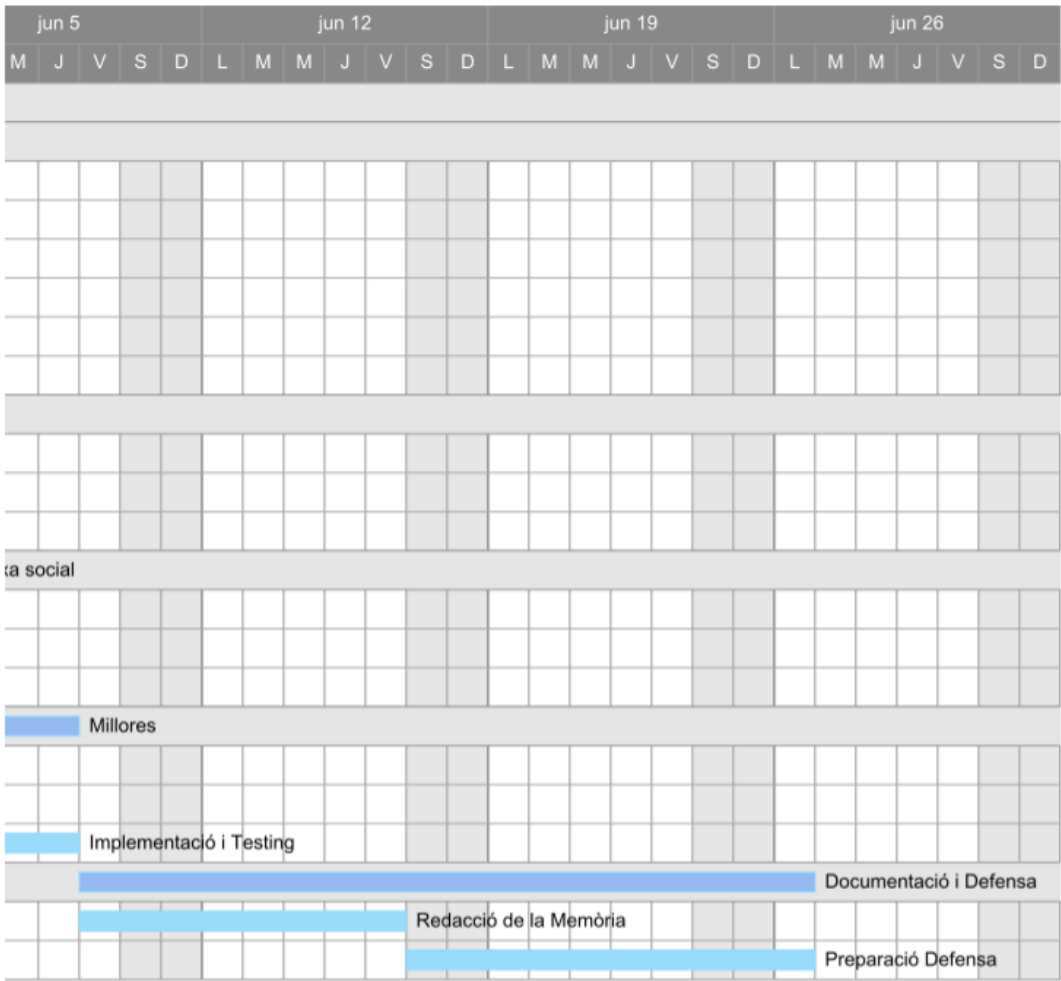
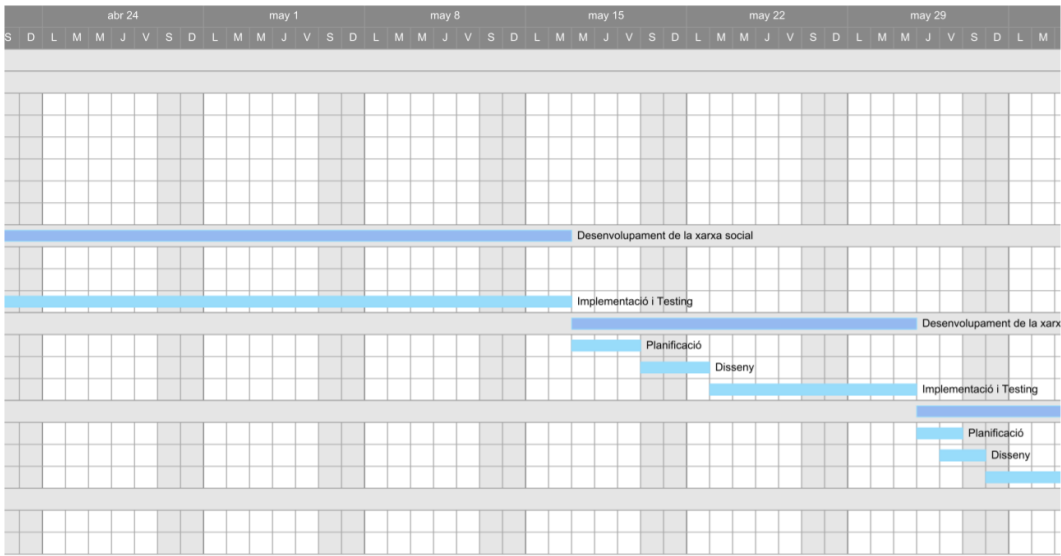


Figura 2.1: Diagrama de Gantt

2.4 Gestió Econòmica

2.4.1. Consideracions inicials

En aquesta secció, identificarem tots els recursos necessaris per a la realització del projecte i farem una estimació del cost total d'aquests, tenint en compte aquestes dues variants:

- Assumim que el projecte és universitari, i que per tant, no té una inversió econòmica.
- Assumim que és un projecte competitiu, amb les hores de recursos humans a preu de mercat.

A més, també s'explicarà com es durà a terme el control de la gestió durant tot el projecte.

2.4.2. Identificació i estimació de costos

Costos directes

Els costos directes per activitat en aquest projecte inclouen els recursos humans involucrats en cada una de les activitats definides al diagrama de Gantt presentat anteriorment. Encara que a la pràctica tot el projecte estigui desenvolupat per una sola persona, cada activitat d'aquest correspon a un rol diferent, i per tant, les hores tindran un preu de mercat diferent:

- Tasques de gestió i documentació (Cap de Projecte): 50€/h.
- Anàlisi i disseny del producte (Analista): 40€/h.
- Implementació i proves (Programador): 35€/h.

A continuació trobem una taula amb el cost de mercat i el cost estimat d'aquest projecte:

Activitat	Unitats	Preu de mercat	Cost de mercat	Preu estimat	Cost estimat
Posada en marxa	20h	50 €/h	1000 €	0 €/h	0 €
GEP	70h	50 €/h	3500 €	0 €/h	0 €
Desenvolupament de la xarxa social - Planificació	100h	50 €/h	5000 €	0 €/h	0 €
Desenvolupament de la xarxa social - Disseny	50h	40 €/h	2000 €	0 €/h	0 €
Desenvolupament de la xarxa social - Implementació i Testing	150h	35 €/h	5250 €	0 €/h	0 €
Desenvolupament de l'eina de gestió - Planificació	20h	50 €/h	1000 €	0 €/h	0 €
Desenvolupament de l'eina de gestió - Disseny	20h	40 €/h	800 €	0 €/h	0 €
Desenvolupament de l'eina de gestió - Implementació i Testing	60h	35 €/h	2100 €	0 €/h	0 €
Milliores - Planificació	10h	50 €/h	500 €	0 €/h	0 €
Milliores - Disseny	10h	40 €/h	400 €	0 €/h	0 €
Milliores - Implementació i Testing	40h	35 €/h	1400 €	0 €/h	0 €
Documentació i Defensa	60h	50 €/h	3000 €	0 €/h	0 €
Total	610h		25950 €		0 €

Taula 2.2: Costos directes

Costos indirectes

En aquesta secció tindrem en compte tots els costos que ens afectin de forma indirecta, ho sigui, considerarem tots els costos que siguin necessaris pel projecte que siguin necessàries per a qualsevol projecte de *software* o que s'hagin de considerar en aquest projecte.

- **Transport:** S'utilitzarà un abonament trimestral de 105 €. Com que l'abonament no depèn del nombre de viatges, considerarem que el cost extra és de 0
- **Software Extern:** S'intentarà fer servir *software* lliure i gratuït, de forma que el seu cost serà 0.
- **Impressions de paper:** el lliurament del projecte, inclou el lliurament de documentació. Suposarem que l'extensió de la memòria és d'unes 150 pàgines. Per altra banda, se'n han de fer 4 còpies: 3 per als membres del tribunal i una pel director del projecte. Comptarem que cada pàgina costa uns 0,05 € (enquadrernació inclosa).
- **Hardware:** L'únic *hardware* que s'utilitzarà en aquest projecte és un ordinador portàtil que ja es troba en ús. Suposant que la vida útil d'un portàtil és de 5 anys i que el 60% del seu ús durant els propers 4 mesos estarà dedicat al projecte.
- **Connexió a internet:** Tot i que la connexió a internet és imprescindible per quasi qualsevol informàtic, considerarem que ja existeix una contractació. A partir d'aquí suposarem que de l'ample de banda consumit, se'n dedica un 20% al projecte durant el temps que aquest duri.

Producte	Unitats	Preu unitari	Percentatge de dedicació	Cost estimat
Transport	4 mesos	35 €/mes	0%	0 €
Software extern	---	---	---	0 €
Impressions de paper	600 pàgines		100%	30 €
Hardware	4 mesos	590 € / 5 anys	60%	23,6 €
Connexió a internet	4 mesos	49,5 €/mes	20%	39,6 €
Total				93,2 €

Taula 2.3: Costos indirectes

Contingència

Per evitar qualsevol risc que hi pugui haver, reservarem una part del pressupost per a la part de contingència. Aquest cost serà un 15% de la suma dels costos directes i indirecte.

Producte	Percentatge	Preu de mercat	Cost de mercat	Preu estimat	Cost estimat
Costos directes	15%	25950 €	3892,5 €	0 €	0 €
Costos indirectes	15%	93,2€	13,98€	93,2€	13,98€
Total			3906,48 €		13,98€

Taula 2.4: Contingència

Imprevistos

Durant el projecte es poden presentar diferents imprevistos que ens facin encarir el projecte, els quals són:

- **Avaria ordinador:** en cas que hi hagués algun problema amb el *hardware*, s'hauria de reparar o comprar-ne un de nou, augmentant el cost del projecte en màxim el preu de l'ordinador nou. Assignarem a aquesta eventualitat una probabilitat del 5%.
- **Retard de 8 dies:** com que en la planificació temporal ja hem fet una reserva de millores del projecte, en cas de que aquest s'allargui s'utilitzarà aquesta reserva per acabar-lo.

Imprevist	Probabilitat	Unitats	Preu de mercat	Cost de mercat	Preu estimat	Cost estimat
Avaria ordinador	30%	0h	40€/h	0 €	0 €	0 €
Retard de 8 dies	5%	1	590 €	29,5 €	590 €	29,5 €
Total				29,5 €		29,5 €

Taula 2.5: Imprevistos

Pressupost

Durant el projecte no es consideren les següents situacions:

- No es tenen en compte els augments de preu (escalació) durant el projecte, ja que es de curta durada i no es preveu que es doni la situació.
- No hi ha cap marge de benefici, ja que el projecte és completament sense ànim de lucre i no està destinat a la venda d'un producte.

Concepte	Cost de mercat	Cost estimat
Costos directes	25950 €	0 €
Costos indirectes	93,2 €	93,2 €
Contingència	3906,48 €	13,98 €
Imprevistos	29,5 €	29,5 €
Total	29979,18 €	136,68 €

Taula 2.6: Pressupost

2.4.3. Control de gestió

Per poder preveure les desviacions a nivell de temps, durant la realització del projecte, es mantindrà un registre d'hores amb data i concepte per tal de veure si es sobrepassa el número d'hores planificat. A més, a l'hora de fer cada una de les iteracions, com ja hem especificat a l'apartat de metodologia i rigor, és durà a terme amb metodologies àgils, cosa que ens donarà bastant marge de planificació en el cas que preveiem que s'hagi d'allargar el projecte, ja que les metodologies àgils ens donen l'opció de fragmentar molt les tasques en tasques més petites.

Al final del projecte, quan ja es tinguin les hores reals que s'han dedicat al projecte i els costos indirectes, es farà una comparació amb la previsió inicial.

En cas de que la comparativa sigui molt extrema es mirarà quines han sigut les causes que han provocat aquesta situació. Si el cost real ha sigut major al pressupostat, es mirarà si ha sigut degut als imprevistos que s'han tingut en compte i si es pot cobrir amb la partida corresponent. Si no es pot, s'assignarà la partida de contingència per cobrir-ho. Només es tindran en compte els costos indirectes, la partida d'imprevistos i la partida de contingència.

2.5 Sostenibilitat i compromís social

En aquest apartat, es farà una breu descripció de com afecta el projecte a nivell econòmic, social i ambiental.

	PPP	Vida útil	Riscs
Ambiental	0	0	0
Econòmic	0	0	0
Social	0	15	-5
Rang sostenibilitat	0	15	-5
	10		

Taula 2.7: Pressupost

2.5.1. Econòmica

Com que el projecte és sense ànim de lucre, he decidit punta'l en 0 punts a nivell econòmic. Ja que no està pensat per fer-hi negoci, sinó a ser una eina que ajudi a integrar a la societat totes aquelles persones que ho necessiten i no tenen l'oportunitat de fer-ho.

2.5.2. Social

L'objectiu principal del projecte, és donar l'opció de poder utilitzar una xarxa social a persones amb diversitat funcional de dèficit cognitiu amb unes nocions mínimes de lectura i escriptura. Per tant, com que el projecte està encarat a fer una ajuda social, he decidit ficar-li 15 punts d'impacte social.

També existeix algun risc social en aquest projecte, ja que pot haver-hi persones que el categoritzin de marginació social pel fet de que només i tinguin accés persones amb diversitat funcional de dèficit cognitiu, o professionals relacionats en l'àmbit. Per això, he decidit ficar-li -5 punts de risc social.

2.5.3. Ambiental

He decidit puntuar tota la secció ambiental amb 0 punts, ja que el projecte no ajuda ni perjudica el medi ambient. A més no és preveu la dependència de cap recurs en acabar el projecte, ni necessitem cap producte nou.

CAPÍTOL 3

Seguiment del Projecte

3.1 Introducció

En aquest capítol, es definirà quin ha sigut el seguiment del projecte i com s'ha dut a terme.

Ja que en el projecte s'utilitzen les metodologies àgils, s'ha definit un *product backlog* inicial amb totes les històries a realitzar durant el projecte. A més, la planificació temporal s'ha dividit en 6 *sprints*:

- 3 *Sprints* pertinents al desenvolupament de la xarxa social.
- 2 *Sprints* pertinents al desenvolupament de l'eina de gestió.
- 1 *Sprint* pertinent a la part de millores.

Abans de realitzar cada un dels *sprints*, es seleccionen les històries del *backlog* que es realitzaran. Quan aquest finalitza, les històries que no s'han acabat es retornen al *backlog*, o en casos molt específics, es descarten del projecte.

En aquesta secció es descriurà quin ha sigut el *backlog* inicial, i com s'ha procedit en cada un dels *sprints*. Cada un dels *sprints* estarà definit pels següents apartats: Resum de l'*sprint*, Descripció i resultat de les històries d'usuari, tecnologies utilitzades (només en els casos en què s'en afegeixi alguna) i conclusions de l'*sprint*.

3.2 Product Backlog

El *product backlog* inicial del projecte ha sigut el següent:

Histories	Punts
Com a usuari, vull poder registrar-me a la xarxa social	5
Com a usuari, vull poder consultar i editar el meu perfil	8
Com a desenvolupador, vull poder consultar un mapa navegacional de totes les pantalles de la xarxa social	5
Com a usuari, vull poder publicar una història al meu mur.	20
Com a desenvolupador vull dissenyar una encriptació de les contrasenyes	5
Com a usuari, vull poder veure els perfils i les històries de les meves amistats.	20
Com a usuari, vull poder loguejar-me a la xarxa social	5
Com a gestor, vull poder veure informació dels usuaris que he convidat	20
Com a gestor, vull poder invitar a usuaris a utilitzar la xarxa social	13
Com a gestor, vull poder crear esdeveniments.	20
Com a usuari, vull poder fer amistat amb altres usuaris.	5
Com a usuari, vull poder buscar usuaris que no tinguin amistat amb mi.	8
Com a usuari, vull poder veure una llista de les sol·licituds d'amistat que tinc.	8
Com a usuari, vull poder veure un llista de les sol·licituds d'amistat que he enviat.	8
Com a usuari, vull poder assistir a esdeveniments.	8
Com a usuari, vull poder veure una llista dels esdeveniments on estic convidat.	8
Com a usuari, vull poder eliminar una història publicada per mi.	5
Com a gestor, vull poder convidar a usuaris a un esdeveniment creat per mi	8
Com a gestor, vull poder veure una llista dels usuaris que he convidat a la xarxa social	10
Com a usuari, vull poder no assistir als esdeveniments	8
Com a usuari, vull poder comentar els esdeveniments.	10
Com a usuari, vull poder enviar missatges provats a altres usuaris	13
Com a usuari, vull poder veure una pantalla amb totes les funcionalitats de la xarxa social.	5
Com a usuari vull poder penjar fotografies arrossegant-les directament al navegador.	20
Com a desenvolupador, vull posar a punt l'entorn de treball	13
Com a usuari, vull que la xarxa social mantingui una estètica adequada	5
Com a desenvolupador, vull crear les entitats que es faran servir en la primera part del projecte	8
Com a desenvolupador, vull que la xarxa social tingui un logotip.	5
Com a desenvolupador, vull fer un disseny previ de les pantalles de la xarxa social	13
Com a usuari, vull que la xarxa social estigui en més d'un idioma	5
Com a usuari vull poder veure els esdeveniments que he assistit i assistiré en un futur a partir d'un calendari.	13
Com a usuari vull poder pujar fotografies a la xarxa social	20
Com a usuari, vull poder consultar un timeline amb totes les publicacions dels altres usuaris amb els quals tinc amistat	10
Com a usuari, vull poder editar una història publicada per mi	13
Com a usuari, vull poder comentar publicacions fetes per altres usuaris.	13
Com a usuari, vull poder donar "Like" a les històries de la xarxa social.	8
TOTAL	363

Taula 3.1: Product Backlog

3.3 Sprint 1

3.3.1. Resum de l'*sprint*

L'*sprint* no ha anat dins dels marges esperats. Dels 64 punts definits en aquest *sprint*, només se n'han aconseguit 31.

Això es deu al poc coneixement de les tecnologies que s'havien d'utilitzar a priori, i que per tant s'ha fet molt difícil a l'hora d'utilitzar-les. També cal dir que moltes d'aquestes tecnologies s'han hagut de buscar en el moment que ha sortit la necessitat d'utilitzar-les i s'han convertit en requisits emergents dins del projecte.

En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat.

3.3.2. Descripció i resultat de les històries d'usuari

Com a desenvolupador, vull posar a punt l'entorn de treball (13 punts)

Aquesta història ha consistit a buscar i instal·lar totes les tecnologies necessàries per posar en marxa el projecte. S'ha dividit en les següents tasques:

- Instal·lar IntelliJ Webstorm.
- Instal·lar Node.js.
- Buscar servei de núvol on puguem penjar la pàgina web.
- Instal·lar els paquets necessaris de Node.js.
- Creació del repositori de GitHub.

Resultat: Totes les tasques que s'han descrit en aquesta història s'han completat en el temps estimat. El problema ha vingut a l'hora de començar a implementar la història Com a usuari, vull poder registrar-me a la xarxa social de l'*sprint* (la qual veurem més endavant), ja que hi ha hagut molts requisits emergents que no s'han tingut en compte anteriorment, i per tant, el temps d'aquesta història ha augmentat moltíssim.

Com a usuari, vull que la xarxa social mantingui una estètica adequada (5 punts)

Aquesta història ha consistit a buscar els colors o fonts (tipus de lletra) perquè la pàgina web tingui una estètica agradable a la vista de l'usuari. S'ha dividit en les següents tasques:

- Buscar els colors primaris.
- Buscar una font.

Resultat: No ha sigut molt difícil trobar els colors primaris i les fonts principals de l'aplicació. Els colors primaris són els següents:



Figura 3.1: Paleta de colors de l'aplicació.

Per troba'ls, he utilitzat una eina que es diu **paletton**, la qual et permet combinar colors de forma senzilla. Per utilitzar-la podeu entrar a la següent URL:

<http://paletton.com/#uid=5300u0kqALsd8VjkyQ8B3GbBuoR>

L'aplicació estarà composta per dues fonts:

- **Montserrat:** És la font que s'utilitzarà en els títols.
- **Raleway:** S'utilitzarà per a la resta de text de l'aplicació.

Com a desenvolupador, vull crear les entitats que es faran servir en la primera part del projecte (8 punts)

Aquesta història ha consistit a crear les entitats de dades que guardarem a la base de dades. S'ha dividit en les següents tasques:

- Creació de l'entitat 'Usuari'
- Creació de l'entitat 'Història'
- Creació de l'entitat 'Esdeveniment'

Resultat: Aquesta història va tenir una molt mala estimació de cost, ja que el temps en què s'ha realitzat ha sigut bastant ràpid. Les entitats resultants són les següents:

Usuari	Història	Esdeveniment
uuid: String name: String surname: String birthday: Date gender: String about: String email: String password: String isAdmin: String foto: String code: String	uuid: String foto: String description: String usuari: String date: Date	uuid: String title: String description: String start: Date end: Date place: String usuari: String

Figura 3.2: Entitats de l'aplicació.

Com a desenvolupador, vull que la xarxa social tingui un logotip (5 punts)

Aquesta història ha consistit en la creació d'un logotip per utilitzar en la nostra aplicació web. S'ha dividit en les següents tasques:

- Buscar Imatges pel logotip.
- Creació del logotip.

Resultat: La tasca s'ha realitzat correctament, i dins de l'estimació donada en punts. Per fer el logotip de l'aplicació, s'han utilitzat les mateixes fonts que s'han triat en la història. Com a usuari, vull que la xarxa social mantingui una estètica adequada d'aquest *sprint*. El resultat és el següent:



Figura 3.3: Logotip de l'aplicació.

Com a desenvolupador, vull poder consultar un mapa navegacional de totes les pantalles de la xarxa social (5 punts)

Resultat: Com hem vist anteriorment, la història Com a desenvolupador, vull posar a punt l'entorn de treball d'aquest *sprint* ens ha fet perdre molt temps. Temps que s'hagués pogut invertir a fer la resta d'històries. Per tant, al veure que no necessitàvem un mapa navegacional en aquest *sprint*, he decidit deixar aquesta història per l'*sprint 2*.

Com a desenvolupador, vull fer un disseny previ de les pantalles de la xarxa social (13 punts)

Resultat: Durant l'*sprint*, a l'hora de ficar-me a fer aquesta història, em vaig adonar que el cost d'aquesta era molt elevat, i que per tant era una pèrdua de temps fer aquest disseny. Comptant que el temps per fer el projecte és molt poc i que el seu valor no és molt elevat pel projecte, he decidit retirar aquesta història.

Com a usuari, vull que la xarxa social estigui en més d'un idioma (5 punts)

Resultat: Vaig estar buscant alguna forma de simplificar l'aplicació i ficar els literals (paraules fixes que es troben a l'aplicació) en un mateix fitxer. La solució que vaig trobar era bastant difícil d'implementar i feia pujar els punts d'aquesta història. Per consegüent, he optat que per cada pàgina de l'aplicació, hi haurà 3 versions (una en anglès, una en castellà i una altra en català). Per tant, la història es retira del *backlog* inicial.

Com a usuari, vull poder registrar-me a la xarxa social (5 punts)

Resultat: La història Com a desenvolupador, vull posar a punt l'entorn de treball ens ha tret molt de temps d'aquest *sprint*, i consegüentment, aquesta història no s'ha pogut acabar i queda per l'*sprint 2*.

Com a usuari, vull poder loguejar-me a la xarxa social (5 punts)

Resultat: La història Com a desenvolupador, vull posar a punt l'entorn de treball ens ha tret molt de temps d'aquest *sprint*, i consegüentment, aquesta història no s'ha pogut ni començar i queda per l'*sprint 2*.

3.3.3. Tecnologies utilitzades

Tecnologies Web

Aquesta tecnologia és necessària per al funcionament de l'aplicació. Si l'aplicació no es troba en el núvol, aquesta no és accessible pels usuaris, i per tant perd el seu valor.

La tecnologia que he triat per utilitzar es diu Heroku (<https://www.heroku.com>), i és una plataforma d'aplicacions al núvol. Aquesta plataforma permet tindre la teva pàgina web en línia les 24h, i a més, té un sistema de storage, on es guarden totes les dades generades pels usuaris de la pàgina.

La URL a la pàgina on es troba l'aplicació d'aquest projecte és:

<https://makeyourfriend.herokuapp.com>

IDE

Com que el projecte és web, també s'ha d'utilitzar una IDE encarada a aquest propòsit. La que he triat per aquest projecte es diu IntelliJ WebStorm (<https://www.jetbrains.com/webstorm>), i és una IDE que està encarada al Javascript, però que també accepta tot el que vindria a ser els llenguatges de frontend, com HTML o CSS.

La llicència d'aquesta IDE, és de pagament, però al ser estudiant de la FIB et surt gratuïta.

Bases de Dades

Com que l'aplicació és una xarxa social, una base de dades relacional no és tan eficient, ja que sempre ha d'estar buscant a les taules. Per consegüent, he decidit utilitzar una base de dades no relacional anomenada neo4j (<https://neo4j.com>). Aquesta base de dades és una graph database que permet crear nodes (on es guarda la informació de l'usuari, històries o esdeveniments) i seguidament relacionar-los entre ells amb una etiqueta.

La seva eficiència en aquest tipus d'aplicacions, cau en què no cal que tinguem una taula amb totes les amistats. Si volem saber quines amistats té un usuari, ens fiquem al node d'aquest usuari i mirem les arestes que contenen el tag "Amistat". Aquestes arestes ens porten directament als nodes de les amistats.

Finalment, ni que en aquest *sprint* no s'hagi fet, segurament serà necessari una base de dades relacional per emmagatzemar les fotografies de l'aplicació i guanyar eficiència. Quan una fotografia es guarda en una base de dades, s'encrpta amb un string (que ocupa molt espai), i per tant, costaria molt fer una cerca per la base de dades.

Framework

El *framework* utilitzat pel projecte és Node.js (<https://nodejs.org>). És un *framework* que et facilita totes les feines encarades a web. Els seus llenguatges bàsics són HTML, CSS i Javascript, tot i que pots afegir les funcionalitats que vulguis instal·lant els paquets necessaris al teu projecte.

Finalment, dir que aquest *framework* et dona molta facilitat per implementar la comunicació client servidor.

Paquets addicionals de Node.js

Com he dit a l'apartat anterior, Node.js et permet instal·lar paquets amb funcionalitats extra al teu projecte. Els paquets que s'ha utilitzat en aquest *sprints* són els següents:

- **body-parser:** Aquest paquet ens permet llegir els camps que ha omplert el client directament des del servidor, sense la necessitat d'haver de fer una crida *ajax* per passar els paràmetres.
- **bootstrap:** És un *framework* per HTML, CSS i Javascript que permet fer les pàgines *responsive* amb facilitat.
- **dotenv:** ens permet carregar fitxers `.env` a l'aplicació.
- **jquery:** És una llibreria de Javascript que et permet escriure el mateix codi de forma més ràpida, i que permet la manipulació del codi HTML de forma senzilla.

- **jquery-validation:** És un *pluguin* de jQuery que serveix per validar els diferents formularis que hi puguin haver a l'aplicació (dates, correus electrònics, contrasenyes, etc).
- **neo4j-driver:** És un *driver* que permet configurar connexions amb Neo4j, ja explicat a l'apartat 3.3. **uuid:** Ens permet auto generar un identificador únic per a cada node de la base de dades.
- **pug:** Aquest paquet permet escriure el codi HTML més ràpidament i permet crear bucles d'informació dinàmica a la pàgina web.
- **font-awesome:** És una llibreria d'icones.
- **w3-css:** Permet fer dissenys més amistosos per l'usuari de l'aplicació amb facilitat.

3.3.4. Conclusions de l'*sprint*

Com podem veure, durant aquest *sprint* s'han fet menys de la meitat dels punts que estaven previstos. Tot hi això, ja tenim quasi totes les tecnologies necessàries instal·lades a l'aplicació i per tant, serà més fàcil utilitzar-les.

Sí que és possible que durant l'*sprint* 2 sorgeixi la necessitat d'alguna tecnologia més, però preveig que seran molt poques.

3.4 *Sprint 2*

3.4.1. Resum de l'*sprint*

L'*sprint* a complert amb els requisits establerts inicialment. Dels 73 punts definits en aquest *sprint*, s'han complert tots.

Com s'ha comentat en l'*sprint* anterior, ficar a punt l'entorn de treball, em va treure molt temps del qual s'havia establert inicialment. Com que l'entorn ja es va ficar a punt en l'*sprint* anterior, en aquest *sprint* no hem tingut el mateix problema, i per tant s'ha pogut complir amb tots els requeriments.

En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat. També es veuran quines han sigut les tecnologies que s'han incorporat al projecte.

3.4.2. Descripció i resultat de les històries d'usuari

Com a usuari, vull poder registrar-me a la xarxa social (5 punts)

Aquesta història ha consistit en la implementació d'una pàgina de registre per la xarxa social. La pàgina de registre conté els següents camps:

- **Nom:** El camp contindrà el nom de l'usuari (o el nom amb el qual vulgui ser reconegut pel sistema).
- **Cognoms:** El camp contindrà els cognoms de l'usuari.
- **Gènere:** En aquest camp s'indicarà si l'usuari és un home o una dona.

- **Data de naixement:** En aquest camp s'informarà la data de naixement de l'usuari.
- **Correu Electrònic:** En aquest camp s'informarà el correu electrònic de l'usuari, el qual servirà com identificador del sistema.
- **Sobre tu:** En aquest camp es pot afegir una breu descripció sobre el mateix usuari.
- **Contrasenya:** El camp es repeteix dues vegades (per assegurar que te'n recordaràs de la contrasenya). La contrasenya haurà de tindre un mínim de 8 caràcters perquè sigui vàlida.
- **Codi d'invitació:** En aquest camp s'haurà d'informar el codi que l'administrador haurà informat amb anterioritat a l'usuari.

Resultat: La història s'ha realitzat amb èxit, tot i que el seu cost estava una mica mal estimat. En veritat ha portat més dels 5 punts estimats inicialment.

Com a usuari, vull poder loguejar-me a la xarxa social (5 punts)

La història ha consistit a poder fer login a l'aplicació informant el correu electrònic i la contrasenya registrats en l'història anterior.

Resultat: La història no ha portat cap problema i s'ha realitzat amb el cost estimat.

Com a desenvolupador vull dissenyar una encriptació de les contrasenyes (5 punts)

Aquesta història s'ha dividit en dues tasques:

- Encriptació de la contrasenya.
- Manutenció de les sessions dels usuaris.

Resultat: La primera tasca de la història s'ha utilitzat la llibreria crypto de Node.js, la qual et permet encriptar una contrasenya en base64 utilitzant l'algoritme "sha256". En la segona tasca s'ha utilitzat la llibreria client-session de Node.js, la qual permet mantenir sessions mitjançant les *cookies* del navegador utilitzat. Els punts de la història s'han estimat adequadament.

Com a usuari, vull poder consultar i editar el meu perfil (8 punts)

La història s'ha dividit en dues parts:

- Oferir l'oportunitat de poder consultar el teu propi perfil.
- Poder editar el teu propi perfil. Aquesta tasca s'ha dividit en dos més:
 - ◊ Poder canviar la teva foto de perfil (o ficar-ne una en cas que no en tinguis).
 - ◊ Poder canviar la descripció del teu usuari (el camp "Sobre tu" que s'informa en el registre d'un usuari).

Resultat: Els punts de la història s'han estimat correctament, i s'ha aconseguit tot el que es demanava. S'ha de dir, que la part de canviar la fotografia de perfil s'ha dut a terme en la història Com a usuari vull poder pujar fotografies a la xarxa social la qual es veurà més endavant.

Com a usuari, vull poder veure una pantalla amb totes les funcionalitats de la xarxa social (5 punts)

Aquesta història ha consistit en crear una pantalla on l'usuari pogués veure amb facilitat totes les funcions de la xarxa social. Les diferents funcions que té són: El meu perfil, les meves històries, els meus amics, els meus esdeveniments, línia temporal, buscar amics, publicar història.

Resultat: La història s'ha estimat de forma adequada i s'ha dut a terme correctament.

Com a desenvolupador, vull poder consultar un mapa navegacional de totes les pantalles de la xarxa social (5 punts)

La història ha consistit en crear un mapa navegacional de les pantalles que pot utilitzar l'usuari del sistema (amb usuari ens referim al fet que no és un administrador de la xarxa), utilitzant nomenclatura UML.

Resultat: La història s'ha estimat de forma adequada i s'ha dut a terme correctament. El resultat és el següent:

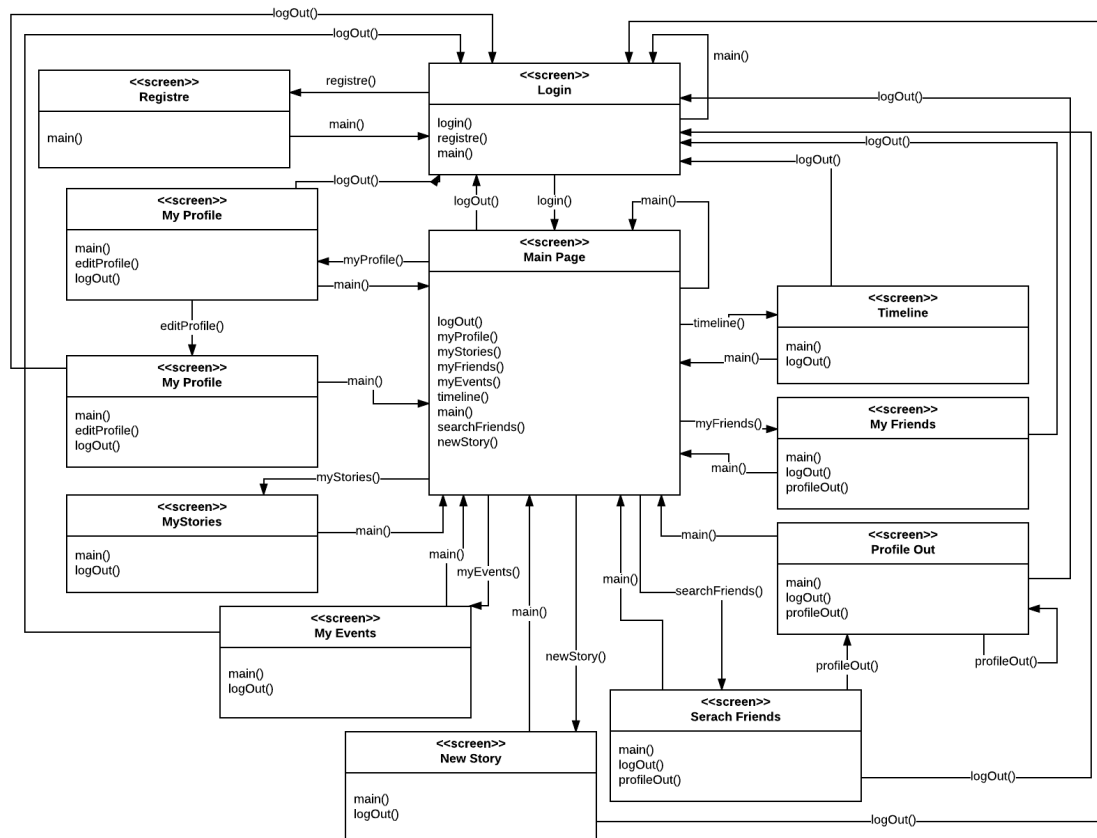


Figura 3.4: Mapa navegacional de l'aplicació (Només usuari).

Com bé podem veure, en aquest mapa navegacional, es pot observar quines seran les diferents pantalles del sistema i com arribar a cada una d'elles. Remarcar, que aquestes pantalles només representen les pantalles accessibles pels usuaris, ja que un administrador tindrà la funcionalitat de poder crear esdeveniments.

Com a usuari, vull poder publicar una història al meu mur (20 punts)

La història ha consistit en donar l'oportunitat a l'usuari per poder publicar històries al seu mur. Per poder fer-ho, s'ha hagut de emplenar un formulari amb els següents camps:

- **Imatge o fotografia:** Et permet penjar una fotografia a la història.
- **Descripció:** Permet afegir una descripció a la història.

Resultat: S'ha dut a terme correctament i s'ha estimat adequadament. Igual que la història anterior, la part de penjar la fotografia s'ha dit a terme en la història. Com a usuari vull poder pujar fotografies a la xarxa social".

Com a usuari vull poder pujar fotografies a la xarxa social (5 punts)

La història ha consistit en poder pujar fotografies a la xarxa social, tant al teu perfil com a les històries que es vulguin publicar.

Resultat: La tasca s'ha dut a terme correctament, tot i que el temps que s'ha utilitzat per completar-la s'ha allargat més del que s'havia previst. Per poder pujar fotografies a la xarxa social he agut d'utilitzar una llibreria de Node.js anomenada multer, la qual et permet pujar fotografies de forma senzilla.

Inicialment, volia fer que es pengessin les fotografies utilitzant un drag & drop des de l'escriptori de l'ordinador fins a la pàgina web. Per fer-ho he trobat moltes complicacions, cosa que ha comportat la creació d'una nova història: Com a usuari vull poder penjar fotografies arrossegant-les directament al navegador".

3.4.3. Tecnologies utilitzades

Bases de Dades

En l'*sprint* anterior, s'havia comentat de què les fotografies s'emmagatzemarien utilitzant una base de dades SQL, tot i que això al final no ha sigut possible.

He estat intentant que en el moment que la imatge arribés al servidor, aquesta es guardés a la base de dades SQL (postgres), però no hi ha hagut manera d'aconseguir-ho, i per tant les imatges es guarden en una jerarquia de directoris al mateix servidor.

Les fotografies de perfil es guardaran en el directori `uploadsProfile` i les imatges de les històries es guardaran en el directori `uploadsPost`. Una vegada les imatges estiguin guardades en el directori corresponent, s'actualitzaran els camps foto de les entitats Usuari i Història esmentades en l'*sprint* anterior.

Paquets addicionals de Node.js

En aquest *sprint* s'han utilitzat els següents paquets addicionals de Node.js:

- **crypto:** Aquest paquet ens permet encriptar les contrasenyes dels usuaris, en base64, utilitzant diferents algoritmes.
- **client-session:** Paquet que ens permetrà mantenir les sessions dels usuaris.

- **multer:** Paquet que permet al client penjar fotografies al servidor utilitzant funcions middleware.

3.4.4. Conclusions de l'*sprint*

Aquest *sprint* s'ha finalitzat correctament i dins del període establert, tot i que també n'ha sortit una nova història, la qual es durà a terme en l'*sprint* 3.

3.5 *Sprint* 3

3.5.1. Resum de l'*sprint*

Durant aquest *sprint*, s'han complert 64 dels 84 punts establerts inicialment, i es pot dir que la primera part del *software* (implementació de la xarxa social) s'ha acabat d'implementar.

En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat. També es veuran quines han sigut les tecnologies que s'han incorporat al projecte.

3.5.2. Descripció i resultat de les històries d'usuari

Com a usuari, vull poder buscar usuaris que no tinguin am estat amb mi (8 punts)

La història ha consistit en poder buscar els usuaris que no tenen am estat amb mi. Aquesta cerca es podia fer de dues formes:

- Buscar tots els usuaris del sistema.
- Buscar a partir d'una cadena de caràcters (els quals coincidissin amb part del nom o cognom de l'usuari).

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder veure un llista de les sol · lituds d'am estat que he enviat (8 punts)

Aquesta història permet a l'usuari poder visualitzar una llista de les sol · lituds d'am estat que té pendants.

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder veure una llista de les sol · lituds d'am estat que tinc (8 punts)

Aquesta història permet a l'usuari poder visualitzar una llista de les sol · lituds d'am estat que aquest ha enviat a altres usuaris.

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder fer amistat amb altres usuaris (5 punts)

La història s'ha centrat en permetre crear amistat amb els usuaris utilitzant part de les històries anteriors que s'han dut a terme en aquest *sprint*.

Resultat: La història s'ha completat amb èxit. Per fer-ho, s'han utilitzat dos tipus de relació entre els usuaris (que es representen en nodes a la base de dades):

- **MY_REQUEST:** Relació d'un usuari cap a un altre que representa que un usuari ha enviat una sol·licitud d'amistat.
- **MY_FRIEND:** Una vegada l'amistat accepta la sol·licitud d'amistat, la relació anterior s'elimina de la base de dades, i seguidament, es creen dues relacions "MY_FRIEND" entre els dos usuaris de la base de dades. Aquestes dues relacions es creen perquè en una graph database no es permet la creació de relacions bidireccionals.

Com a usuari, vull poder veure els perfils i les històries de les meves amistats (20 punts)

Aquesta història ha constat de dues parts:

- Poder visualitzar una llista amb totes les amistats de l'usuari.
- Poder veure els perfils i les històries de les meves amistats.

Resultat: La història s'ha completat amb èxit.

La primera part, ha estat relativament senzilla, ja que part del desenvolupament ja s'havia dut a terme en les històries anteriors.

La segona part, ha costat una mica més, però també ha estat relativament senzill, ja que tenia la plantilla del perfil de l'*sprint* anterior.

Com a usuari, vull poder eliminar una història publicada per mi (5 punts)

Aquesta història consistia en donar l'oportunitat d'esborrar una història publicada anteriorment.

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder consultar un timeline amb totes les publicacions dels altres usuaris amb els que tinc amistat (10 punts)

Aquesta història consistia en poder veure un timeline (ordenat per data de publicació descendiment) de totes les històries publicades per les amistats de l'usuari.

Resultat: La història s'ha completat amb èxit. A més, en aquest punt del *software* ja teníem una pantalla on es permetia visualitzar totes les històries d'un mateix usuari, i per tant només ha faltat fer una nova cerca a la base de dades.

Com a usuari vull poder penjar fotografies arrossegant-les directament al navegador (20 punts)

La història consistia en poder penjar fotografies a la pàgina web utilitzant un drag & drop des de l'escriptori al navegador.

Resultat: La història no s'ha pogut dur a terme, ja que durant aquest *sprint* he tingut moltes coses a fer, i per tant, he decidit deixar aquesta història per l'*sprint* 6, corresponent a la part de millores.

3.5.3. Conclusions de l'*sprint*

Tot i que no s'han complert tots els punts d'història d'aquest *sprint*, les funcionalitats que es demanaven per aquesta part del projecte s'han complert amb èxit, i per tant podem donar la implementació de la xarxa social per finalitzada.

3.6 *Sprint* 4

3.6.1. Resum de l'*sprint*

L'*sprint* 4 és el primer dels dos *sprints* en què es durà a terme l'eina de gestió, la qual consistirà en crear esdeveniments i fer que els usuaris del sistema i puguin assistir.

Dels 38 punts establerts inicialment, s'han pogut complir tots, i per tant, podem dir que s'ha complert amb els requisits de l'*sprint*.

Com podem veure, els punts que s'han assignat a aquest *sprint* són menors que en els anteriors. Això es deu al fet que la duració dels dos *sprints* planificats per desenvolupar aquesta eina són més curts, i per tant no hi ha tant temps com en els anteriors.

En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat. També es veuran quines han sigut les tecnologies que s'han incorporat al projecte.

3.6.2. Descripció i resultat de les històries d'usuari

Com a gestor, vull poder crear esdeveniments (20 punts)

La història ha consistit en poder crear esdeveniments a partir d'un formulari amb els següents camps:

- **Títol de l'esdeveniment:** Contindrà al títol de l'esdeveniment.
- **Data d'inici:** Contindrà la data i l'hora d'inici de l'esdeveniment.
- **Data de finalització:** Contindrà la data i l'hora de finalització de l'esdeveniment.
- **Lloc:** Contindrà el lloc on es durà a terme l'esdeveniment.
- **Descripció:** Contindrà la descripció de les diferents activitats que es realitzaran durant l'esdeveniment.

Cal dir, que aquesta funcionalitat estarà limitada només als administradors del sistema.

Resultat: La història s'ha completat amb èxit.

Com a gestor, vull poder convidar a usuaris a un esdeveniment creat per mi (8 punts)

La història consistia en poder invitar les teves amistats als diferents esdeveniments, una vegada que aquestes estan creats.

Resultat: La història s'ha completat amb èxit. Per fer-ho s'ha utilitzat part de la història *Com a usuari, vull poder buscar usuaris que no tinguin amistat amb mi* desenvolupada en l'*sprint* anterior.

Com a gestor, vull poder veure una llista dels usuaris que he convidat a la xarxa social (8 punts)

La història consistia en poder veure una llista dels usuaris que un gestor havia convidat a utilitzar la xarxa social.

Resultat: La història s'ha completat amb èxit.

3.6.3. Conclusions de l'*sprint*

L'*sprint* s'ha dut a terme dins dels límits establerts i, com s'ha pogut veure en el resum de l'*sprint*, s'han complert tots els punts que es demanaven.

3.7 Sprint 5 ---

3.7.1. Resum de l'*sprint*

Aquest *sprint* es correspon al segon dels dos *sprints* planificats pel desenvolupament de l'eina de gestió.

Durant aquest *sprint*, s'acabarà d'implementar la gestió dels esdeveniments i farà alguna millora del *software*.

En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat. També es veuran quines han sigut les tecnologies que s'han incorporat al projecte.

3.7.2. Descripció i resultat de les històries d'usuari

Com a usuari, vull poder veure una llista dels esdeveniments on estic convidat (8 punts)

Aquesta història ha consistit en donar l'oportunitat de desenvolupar una llista dels esdeveniments on estic convidat.

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder assistir a esdeveniments (8 punts)

Utilitzant la llista implementada en la història anterior "Com a usuari, vull poder veure una llista dels esdeveniments on estic convidat", aquesta història dóna l'oportunitat d'acceptar la invitació i poder assistir a l'esdeveniment.

Resultat: La història s'ha completat amb èxit.

Com a usuari, vull poder no assistir als esdeveniments (8 punts)

Utilitzant la llista implementada en la història "Com a usuari, vull poder veure una llista dels esdeveniments on estic convidat", aquesta història dóna l'oportunitat de no assistir a l'esdeveniment. A més, si anteriorment ja has marcat que volies assistir, també et dóna l'opció de cancel·lar l'assistència a l'esdeveniment.

Resultat: La història s'ha completat amb èxit.

Com a usuari vull poder veure els esdeveniments que he assistit i assistiré en un futur a partir d'un calendari (13 punts)

Aquesta història permet visualitzar un calendari amb tots els esdeveniments que un usuari ha assistit ho assistirà.

Resultat: La història s'ha completat amb èxit. Per dur a terme aquesta història s'han hagut d'incorporar dues llibreries més de Node.js.

Com a usuari, vull poder donar "M'agrada" a les històries de la xarxa social (8 punts)

Aquesta història es correspon a la primera millora del sistema, i permet als usuaris indicar que els hi agrada una història (tan seva com d'un altre usuari).

La història s'ha dividit en dues parts:

- Donar l'oportunitat d'indicar que t'agrada una història de la xarxa.
- Les històries de la xarxa mostren el nombre de reaccions que han tingut.

Resultat: La història s'ha completat amb èxit.

3.7.3. Tecnologies utilitzades**Paquets addicionals de Node.js**

En aquest *sprint* s'han incorporat els següents paquets de Node.js:

- **moment:** Permet definir formats de dates.
- **fullcalendar:** Permet visualitzar diferents esdeveniments en un calendari a partir d'un JSON.

3.7.4. Conclusions de l'*sprint*

L'*sprint* s'ha dut a terme dins dels límits establerts i, com s'ha pogut veure en el resum de l'*sprint*, s'han complert tots els punts que es demanaven. A més, encara que no escaigui dins d'aquest *sprint*, s'ha començat amb la primera millora del sistema: Com a usuari, vull poder donar "M'agrada" a les històries de la xarxa social.

3.8 *Sprint* 6

3.8.1. Resum de l'*sprint*

Aquest *sprint* es correspon a l'últim *sprint* d'implementació de l'aplicació. Dels 53 punts establerts, s'han complert tots.

Durant aquest *sprint*, es durant a terme vàries tasques de millora del *software* i es finalitzaran alguna de les històries que no s'han pogut dur a terme en els *sprints* anteriors. Addicionalment, durant aquest *sprint* s'han resolt una sèrie de *bugs* que s'han trobat a l'hora de testejar el *software*. En els apartats següents veurem quines són les tasques que s'han dut a terme en aquest *sprint* i quin ha sigut el seu resultat. També es veuran quines han sigut les tecnologies que s'han incorporat al projecte.

3.8.2. Descripció i resultat de les històries d'usuari

Com a gestor, vull poder invitar a usuaris a utilitzar la xarxa social (13 punts)

Aquesta història ha consistit en què el gestor/administrador de l'aplicació pugui invitar a nous usuaris a utilitzar la xarxa mitjançant un codi.

Resultat: La història s'ha completat amb èxit. Per fer-ho, s'agafa l'identificador de la base de dades del gestor (el qual ens servirà com a codi d'invitació). Aquest, informa els nous usuaris el codi i finalment, els usuaris afegeixen el codi en el nou camp que s'ha creat a la finestra de registre.

Com a gestor, vull poder veure informació dels usuaris que he convidat (20 punts)

Aquesta història ha consistit en poder visualitzar de forma fàcil tots els usuaris que es trobaven dins de la xarxa social a partir del codi d'invitació.

Resultat: La història s'ha completat amb èxit. Utilitzant els codis que els usuaris introdueixen a l'hora de registrar-se, es pot saber fàcilment, quin administrador l'ha invitat.

Com a usuari vull poder penjar fotografies arrossegant-les directament al navegador (20 punts)

Aquesta història consistia en poder fer drag & drop d'una imatge des d'una finestra de la pantalla fins al navegador directament.

Resultat: La història s'ha completat amb èxit. Tot i que ha portat molts problemes, finalment s'ha pogut solucionar.

3.8.3. Tecnologies utilitzades

Paquets addicionals de Node.js

En aquest *sprint* s'han incorporat un nou paquet de Node.js. Aquest paquet s'anomena *materialize-css*, i ens ha servit per poder retocar la interfície gràfica i fer-la una mica més amigable.

3.8.4. Conclusions de l'*sprint*

En aquest *sprint* no podem dir el mateix que els anteriors, ja que no s'ha pogut dur a terme dins dels límits establerts. Segons la planificació inicial, l'*sprint* s'hauria d'haver finalitzat el 8 de Juny, però fins al dia 16 no ha finalitzat.

Aquest fet ha implicat que hi hagués una superposició entre les tasques "Millores i Documentació i defensa", definides en l'apartat 2.3.2 del document.

Tot i que hi ha hagut una superposició de les tasques, s'han dut a terme de forma paral·lela, i per tant, la planificació s'ha tornat a encarrilar al ritme que toca.

3.9 Conclusions de Seguiment

S'han dut a terme tots els *sprints*, tot i que, com s'ha comentat en la secció anterior, l'últim *sprint* no s'ha dut a terme dins dels marges establerts.

Adicionalment, hi ha hagut 4 històries del *backlog* que no s'han pogut dur a terme per falta de temps:

- Com a usuari, vull poder comentar els esdeveniments. (10 punts)
- Com a usuari, vull poder enviar missatges provats a altres usuaris. (13 punts)
- Com a usuari, vull poder editar una història publicada per mi. (13 punts)
- Com a usuari, vull poder comentar publicacions fetes per altres usuaris. (13 punts)

Tot i els inconvenients que hi ha hagut, podem dir que el *software* final és un *software* usable i que conté totes les funcionalitats essencials establertes a l'inici del projecte.

CAPÍTOL 4

Informació Tècnica

4.1 Introducció

En aquest capítol, explicaré com s'ha dissenyat el *software*, com s'ha implementat i quines proves s'han fet per validar el *software*. D'aquesta forma, que en el moment que algú vulgui continuar amb el desenvolupament del *software*, no tingui cap problema.

La versió més actualitzada del còdi es troba en el següent repositori de GitHub:

<https://github.com/daleu/makeyourfriend>

Addicionalment, l'última versió de l'aplicació es troba desplegada a la següent URL:

<https://makeyourfriend.herokuapp.com>

Per poder crear un usuari en aquesta pàgina, s'haurà d'introduir el següent codi de validació:

3818207-5651-416d-ac34-e49228d6ee7f

I les credencials per poder accedir amb administrador són:

email: david.aleu.16@gmail.com

password: 11111111

4.2 Disseny

4.2.1. Arquitectura

L'aplicació s'estructurarà seguint l'arquitectura en tres capes. Concretament utilitzarem la capa de presentació, la capa de negoci i la capa de dades. D'aquesta forma, aconseguim tenir les diferents funcionalitats del sistema separades pel tipus d'objectiu que tenen.

Seguidament, descriurem els objectius de cada una de les capes de forma breu.

Capa de Presentació

Aquesta capa és la responsable de la interacció amb l'usuari.

Per exemple, anem a suposar que un usuari vol crear una nova història. Aquest introduirà la descripció de la història, i opcionalment, una fotografia. Seguidament clicarà el

botó de publicar història. Les dades que ha introduït l'usuari s'enviaran a la capa de negoci.

La capa de presentació, no serveix només perquè l'usuari pugui introduir dades. En aquesta capa també es mostren les dades que conté l'aplicació a l'usuari, de forma que aquestes es puguin entendre amb facilitat.

Els fitxers del projecte que formen aquesta capa, són tots els fitxers que contenen la carpeta `/views` (on es troben totes les pantalles de l'aplicació) i la carpeta `/public` (on es defineixen estils i funcions comunes de les vistes). Part de la capa de presentació, es troba en la carpeta `/node_modules`.

Capa de Negoci

La capa de negoci, és la capa que conté tota la lògica del sistema: transformació de dades, manteniment de les sessions d'usuari, transferència de dades entre la capa de dades i la capa de presentació, execució d'algoritmes del sistema, etc.

Seguint l'exemple anterior, una vegada les dades de la nova història arriben a la capa de negoci, aquestes es transformen de forma que siguin acceptables per a la capa de dades. Addicionalment, s'agafa el moment en què s'ha produït la nova història, i l'usuari que l'ha fet. Una vegada s'han fet totes les transformacions pertinents, s'envia a la capa de dades.

L'únic fitxer del projecte que forma aquesta capa és el fitxer `app.js`. En aquest fitxer, es troben totes les rutes de la capa de presentació i totes les interaccions amb la capa de dades. A part, també es defineix totes les variables que permeten configurar l'entorn.

Capa de Dades

La capa de dades, és la capa responsable de comunicar-se amb la base de dades i que manté la persistència del sistema. Guarda totes les dades que li arriben de la capa de negoci i serveix totes les dades que aquesta li pugui demanar.

Continuant amb l'exemple anterior, una vegada les dades de la nova història arriben a la capa de dades, aquesta serà la responsable de fer la *query* a la base de dades per inserir-hi les noves dades.

L'únic fitxer que forma la capa de dades, és el fitxer `neo4j-api.js` i és el responsable d'interactuar amb la base de dades.

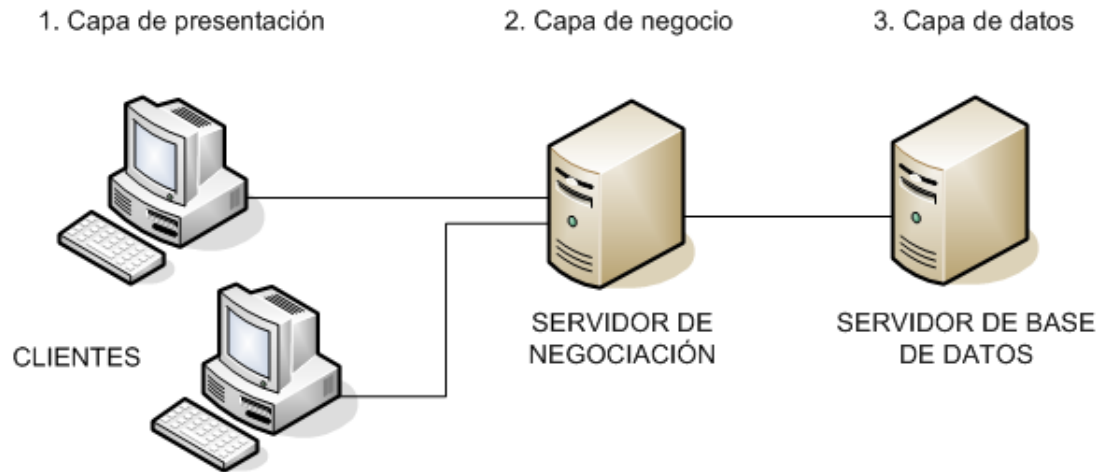


Figura 4.1: Arquitectura 3 capes.

4.2.2. Tecnologia utilitzada

Per la implementació del *software*, hem utilitzat una sèrie de tecnologies, *frameworks* i llenguatges de programació. Aquestes tecnologies es llisten a continuació.

HTML5

HTML, o Hyper Text Markup Language, és el llenguatge bàsic que s'utilitza per estructurar documents web. És un llenguatge d'etiquetes que permet definir diferents elements en una pàgina web. Aquests elements poden ser botons, enllaços, imatges, text pla, o qualsevol tipus d'element que ens ajudi a crear una pàgina web.

CSS

CSS, o Cascading Style Sheets, és un llenguatge de fulls d'estil que permet descriure la presentació (tant estil com format) d'un document basat en llenguatge d'etiquetes.

Javascript

Javascript és un llenguatge basat en prototipus. Aquest llenguatge, juntament amb la llibreria jQuery, ens serveixen per manipular elements de la pàgina web, o l'estil d'aquests, de forma dinàmica.

Neo4j

Neo4j és una base de dades no relacional, la qual es caracteritza per no tenir un esquema definit. És una graph database, el qual implica que la base de dades estarà definida per nodes i relacions entre aquests.

En una aplicació com la que s'està implementant ens és molt útil, ja que la informació d'aquesta va fluctuant tota l'estona i les cerques siguin més ràpides, ja que totes surten d'un node concret.

JSON

JSON, o Javascript Object Notation, és un llenguatge obert basat en text i orientat a l'intercanvi de dades. A més és fàcil d'entendre per qualsevol persona, inclús si aquesta no es troba en el món de la informàtica.

Per agafar la informació de la base de dades, necessitem implementar una API o un connector amb Javascript, de forma que aquest ens permeti tenir un flux d'informació. Per fer-ho, totes les queries de consulta que es fan retornen el seu resultat en format JSON.

Node.js

És un *framework* que et facilita totes les feines encarades a web. Els seus llenguatges bàsics són HTML, CSS i Javascript, tot i que pots afegir les funcionalitats que vulguis instal·lant els paquets necessaris al teu projecte.

Finalment, dir que aquest *framework* et dona molta facilitat per implementar la comunicació client servidor.

Els paquets addicionals que s'han utilitzat en l'aplicació són els següents:

- **body-parser:** Aquest paquet ens permet llegir els camps que ha omplert el client directament des del servidor, sense la necessitat d'haver de fer una crida *ajax* per passar els paràmetres.
- **bootstrap:** És un *framework* per HTML, CSS i Javascript que permet fer les pàgines *responsive* amb facilitat.
- **dotenv:** ens permet carregar fitxers *.env* a l'aplicació.
- **jquery:** És una llibreria de Javascript que et permet escriure el mateix codi de forma més ràpida, i que permet la manipulació del codi HTML de forma senzilla.
- **jquery-validation:** És un *pluguin* de jQuery que serveix per validar els diferents formularis que hi puguin haver a l'aplicació (dates, correus electrònics, contrasenyes, etc.).
- **neo4j-driver:** És un *driver* que permet configurar connexions amb Neo4j, ja explicat a l'apartat 3.3. **uuid:** Ens permet auto generar un identificador únic per a cada node de la base de dades.
- **pug:** Aquest paquet permet escriure el codi HTML més ràpidament i permet crear bucles d'informació dinàmica a la pàgina web.
- **font-awesome:** És una llibreria d'icones.
- **w3-css:** Permet fer dissenys més amistosos per l'usuari de l'aplicació amb facilitat.
- **crypto:** Aquest paquet ens permet encriptar les contrasenyes dels usuaris, en base 64, utilitzant diferents algorismes.
- **client-session:** Paquet que ens permetrà mantenir les sessions dels usuaris.
- **multer:** Paquet que permet al client penjar fotografies al servidor utilitzant funcions *middleware*.

- **moment:** Permet definir formats de dates.
- **fullcalendar:** Permet visualitzar diferents esdeveniments en un calendari a partir d'un JSON.
- **materialize-css:** Paquet que m'ha permès fer la interfície gràfica més amigable. Aquest paquet, conté una sèrie de classes css i funcions javascript que permeten tenir una millor interacció amb la UI de l'aplicació.

Webstorm

Webstorm és una IDE (Integrated development environment), creada per JetBrains, encarada al desenvolupament d'aplicacions web i que ens servirà per fer la implementació de l'aplicació.

Els seus llenguatges principals són Javascript, HTML i CSS.

4.2.3. User Interface

Fonts

L'aplicació estarà composta per dues fonts:

- **Montserrat:** És la font que s'utilitzarà en els títols.
- **Raleway:** S'utilitzarà per a la resta de text de l'aplicació.

He triat aquestes fonts a partir de les guies d'estil de google (<https://fonts.google.com/>). Montserrat és la font que ens servirà per als títols, i Raleway com a font sans-serif, que és la que es recomana per la major part del text de la pàgina.

Colors primaris de l'aplicació

L'aplicació està composta per la següent paleta de colors:



Figura 4.2: Paleta de colors de l'aplicació.

Per troba'ls, he utilitzat una eina que es diu **paletton**, la qual et permet combinar colors de forma senzilla. Per utilitzar-la podeu entrar a la següent URL:

<http://paletton.com/#uid=5300u0kqALsd8VjkyQ8B3GbBuoR>

Si us fixeu, en quasi totes les xares socials conegudes, s'utilitza el blau com a color principal de l'aplicació. Jo he decidit ficar el ver per trencar una mica els esquemes i que l'aplicació pugui ser fàcilment reconeguda.

Logotip

Per fer el logotip de l'aplicació, s'han utilitzat les mateixes fonts que s'han esmentat anteriorment. El resultat és el següent:



Figura 4.3: Logotip de l'aplicació.

El logotip, intenta representar una mà com a símbol representatiu de l'amistat. D'aquí el nom de MakeYourFriend.

4.2.4. Mapes navegacionals

A continuació, es mostren els dos mapes navegacionals, tant pels usuaris com pels administradors del sistema:

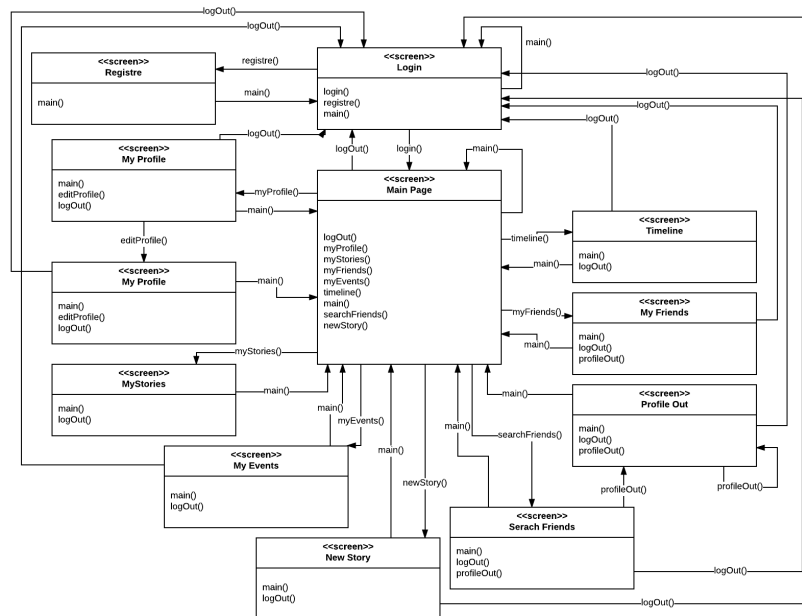


Figura 4.4: Mapa navegacional de l'aplicació (Usuari).

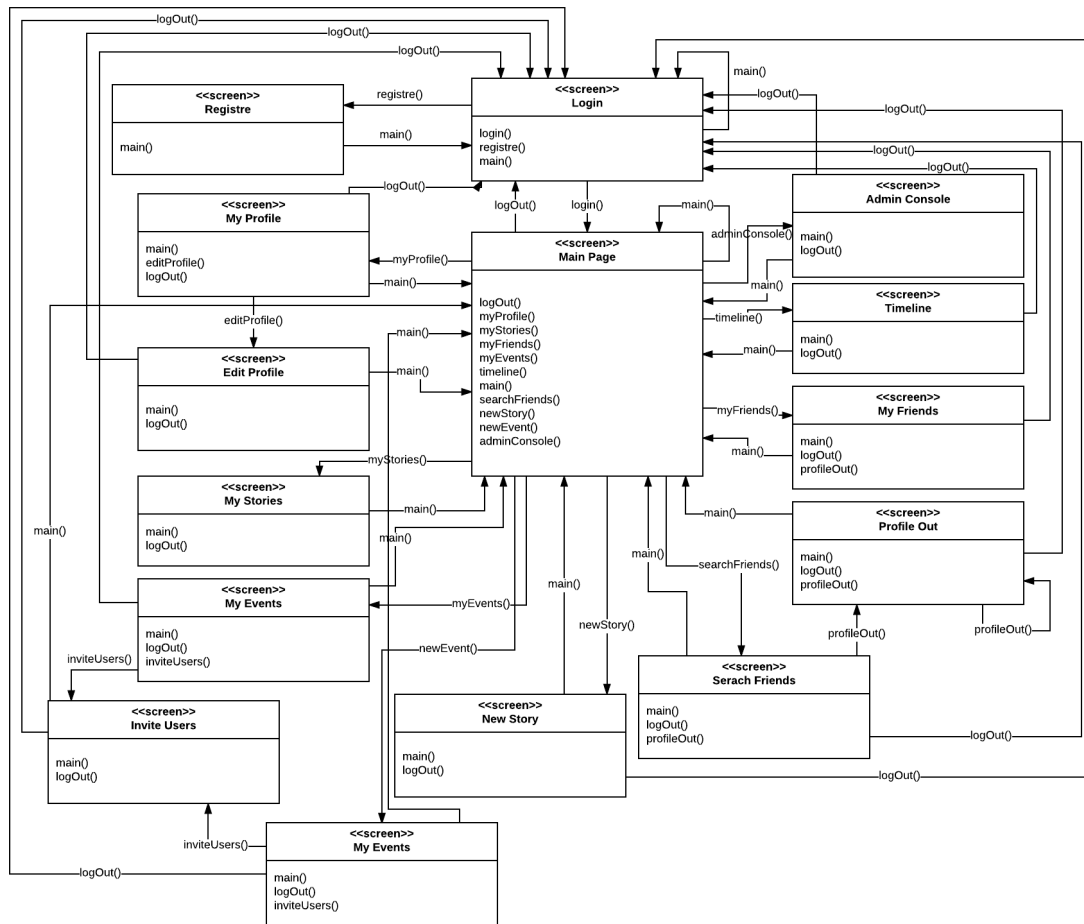


Figura 4.5: Mapa navegacional de l'aplicació (Gestor/Administrador).

4.2.5. Model de dades

Com hem dit anteriorment, en la base de dades que s'utilitza no hi ha un esquema definit. Tot hi això, s'han definit unes entitats i unes relacions entre aquestes per mantenir una constància de la informació. Les entitats són les següents:

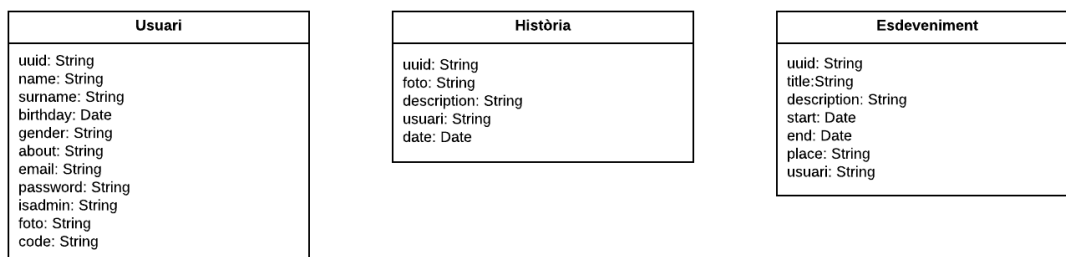


Figura 4.6: Entitats de l'aplicació.

Usuari

L'usuari representa tant els usuaris del sistema com el seu gestor/administrador. Els camps que té aquesta entitat són els següents:

- **uuid:** Representa l'identificador únic dins de la base de dades.
- **name:** Conté el nom de l'usuari.
- **surname:** Conté el cognom o cognoms de l'usuari.
- **birthday:** Conté la data de naixement de l'usuari.
- **gender:** Conté el gènere de l'usuari. El gènere pot ser "Male" o "Female".
- **about:** Conté la descripció de l'usuari. El que l'usuari vol que sàpiguen d'ell.
- **email:** Conté el correu electrònic de l'usuari.
- **password:** Conté la contrasenya en base-64 de l'usuari.
- **isAdmin:** Camp que informa de si l'usuari és administrador/gestor.
- **foto:** Conté la ruta on està guardada la fotografia del seu perfil.
- **code:** Conté el codi d'administrador. Els usuaris necessiten un codi per poder registrar-se al sistema, i aquest camp conté el codi, el qual es correspon amb el uuid de l'administrador.

Història

Una història representa una publicació que l'usuari a fet dins del sistema. Aquesta publicació es guarda amb els següents camps:

- **uuid:** Representa l'identificador únic dins de la base de dades.
- **foto:** Conté la ruta de la fotografia que l'usuari ha afegit a la història. En cas que la història no tingui cap foto, el camp es defineix amb "nothing".
- **description:** Conté la descripció de la història.
- **usuari:** Conté el nom i el cognom de l'usuari que he fet la publicació.
- **date:** conté la data en què s'ha creat la història.

Esdeveniment

Un esdeveniment és una entitat en què els usuaris poden indicar si i volen assistir o no. Aquest esdeveniment conté els següents camps:

- **uuid:** Representa l'identificador únic dins de la base de dades.
- **title:** Conté el títol de l'esdeveniment.
- **description:** Conté la descripció de l'esdeveniment.
- **start:** Conté el dia i l'hora d'inici de l'esdeveniment.

- **end:** Conté el dia i l'hora de finalització de l'esdeveniment.
- **place:** Conté el lloc on es durà a terme l'esdeveniment.
- **usuari:** Conte el nom i el cognom de l'usuari que ha creat l'esdeveniment.

Tot i tenir les entitats, no podem contenir totes les dades del sistema, ja que aquestes entitats només contenen informació d'una unitat única dins de l'aplicació. Per això, s'han definit diferents relacions entre les entitats del sistema. Perquè es puguin entendre les diferents relacions, s'utilitzarà el llenguatge Cypher, que és el llenguatge de queries que s'utilitza en les bases de dades neo4j. Les relacions es definiran de la següent manera:

- **(exemple):** Els parèntesis indiquen que és un node, i exemple representaria el nom del node.
- **-[exemple]->;** `-[]->`; representa una relació d'un node a un altre, i exemple seria el nom de la relació.

Les relacions que tenim dins del sistema són les següents:

- **(Usuari A)-[MY_FRIEND]->;(Usuari B):** Aquesta relació representa que l'usuari A té amistat amb l'usuari B. Com que neo4j no permet una relació bidireccional, si en el sistema trobem una relació MY_FRIEND d'un node a un altre, implica que existirà la mateixa relació en sentit contrari.
- **(Usuari A)-[MY_REQUEST]->;(Usuari B):** Aquesta relació representa que l'usuari A ha sol·licitat tenir amistat amb l'usuari B. Si existeix aquesta relació, implica que l'usuari A no pot tenir una relació de MY_FRIEND (explicada anteriorment) amb l'usuari B i a la inversa.
- **(Usuari A)-[MY_POST]->;(Post A):** Aquesta relació indica que el Post A (Història A) ha estat creada per l'usuari A.
- **(Usuari A)-[LIKE]->;(Post A):** Aquesta relació indica que a l'usuari A li agrada el Post A (Història A).
- **(Usuari A)-[MY_EVENT]->;(Esdeveniment A):** Aquesta relació indica que l'esdeveniment A ha estat creat per l'usuari A.
- **(Esdeveniment A)-[EVENT_REQUEST]->;(Usuari A):** Aquesta relació indica que l'usuari A està convidat a l'esdeveniment A.
- **(Usuari A)-[ASSIST_EVENT]->;(Esdeveniment A):** Aquesta relació indica que l'usuari A assistirà a l'esdeveniment A.

3. Una vegada la *cookie* està configurada, aquesta s'ha de passar al client en el moment que fa login:

```
function setLogin(email, password, res, req, page, pageer) {
  db.findUser(email).then((usr) => {
    if (!usr[0]) {
      res.render(pageer, { error: 'Invalid email or password.' });
    } else {
      const hash = crypto.createHash('sha256').update(password).digest('base64');
      if (usr[0].password == hash) {
        user = usr[0];
        req.session.user = user;
        res.redirect(page);
      } else {
        res.render(pageer, { error: 'Invalid email or password.' });
      }
    }
  });
}
```

Com podem veure, en primer lloc validem que l'usuari i la contrasenya siguin vàlids. Per fer-ho ens comuniquem amb la capa de dades (db en el codi), mirem si l'usuari existeix, i si és així, comprovem la seva contrasenya. En el cas que no sigui vàlid, redireccionem al login amb error, i en cas que sigui correcte el redireccionem a la pàgina principal i li passem la *cookie* amb tota la seva informació.

4. Una vegada l'usuari té la *cookie*, per cada crida que fa al servidor s'executa la següent funció:

```
function requireLogin(req, res, next) {
  if (!req.session.user) {
    if (req.get('accept-language').includes('ca')) {
      res.redirect('/login-ca');
    } else if (req.get('accept-language').includes('es')) {
      res.redirect('/login-es');
    } else res.redirect('/login-en');
  } else {
    next();
  }
}
```

En el cas que la *cookie* ja no sigui vàlida (tant per què la *cookie* ha caducat, com que s'ha tancat el navegador), es redirecciona a la pàgina d'inici de l'aplicació (el login). Altrament, navegarà a la pàgina que l'usuari hagi demanat.

Tot el codi d'aquest apartat, es troba en el fitxer `app.js`.

4.3.2. Registre

En aquest apartat, es donarà com a exemple com s'ha fet el registre d'un nou usuari. Per fer-ho, s'han fet servir uns quants fitxers, els quals s'aniran anomenant pas a pas:

1. Per poder registrar un usuari al sistema, s'ha d'omplir el següent formulari, corresponent al fitxer que es troba en la ruta /views/register/register-en:

Figura 4.8: Pàgina de registre.

2. Una vegada s'omple el formulari i es clica el botó register, s'executa el següent codi (que es troba en el mateix fitxer):

```

$( "#submitbutton" ).click(function() {
    var form = $("#registerform");
    form.validate({
        rules: {
            email: {
                required: true,
                email: true
            },
            pass1: {
                required: true,
                minlength: 8
            },
            pass2: {
                required: true,
                equalTo: "#pass1"
            },
            birthday:{
                isValidDate: true
            }
        }
    });

    if(form.valid()) {
        $('#spinnerbackground').css('display', 'block');
        $('#spinner').css('display', 'block');
        $.ajax({
            url: "/getUsers"
        }).done(function (users) {
            var exists = false;

```



```

    var validCode = false;
    for(i=0;i<users.length;++i){
        if(users[i].email == document.getElementsByName("
            email")[0].value){
            exists = true;
            i=users.length;
        }
        if(users[i].uuid.toLowerCase() == document.
            getElementsByName("code")[0].value.toLowerCase
            () && users[i].isadmin=="YES"){
            validCode = true;
        }
    }
    if (exists) {
        $('#spinnerbackground').css('display', 'none');
        $('#spinner').css('display', 'none');
        $('#infobackground').css('display', 'block');
        $('#errorcard').css('display', 'block');
    }
    else {
        if(validCode == false){
            $('#spinnerbackground').css('display', 'none');
            $('#spinner').css('display', 'none');
            $('#infobackground').css('display', 'block');
            $('#errorcardcode').css('display', 'block');
        }
        else form.submit();
    }
    }).fail(function () {
        $('#spinnerbackground').css('display', 'none');
        $('#spinner').css('display', 'none');
    });
}
});

```

Com podem veure, en primer lloc, es valida que el formulari s'hagi omplert correctament. Seguidament s'executa una *query* per agafar tots els usuaris del sistema. Aquesta *query*, anirà al fitxer app.js (capa de negoci) i seguidament al fitxer neo4j-api.js (capa de dades), Agafarà els usuaris del sistema i els mostrarà al client. A partir d'aquí, es validarà que el codi de validació introduït per l'usuari sigui vàlid i que l'usuari no existeixi al sistema.

3. Si el codi de validació no és vàlid, l'usuari haurà de tornar d'omplir el camp de codi d'usuari. Altrament, es farà submit del formulari i s'executarà la petició post de /registre-en, en l'arxiu app.js (el qual conté totes les rutes de l'aplicació.):

```

app.post('/registre-en', (req, res) => {
    const name = req.body.name;
    const surname = req.body.surname;
    const gender = req.body.gender;
    const code = req.body.code;

    // FORMAT TO STANDAR DATE
    const auxDate = req.body.birthday;
    const parts = auxDate.split('/');
    const birthday = `${parts[1]}/${parts[0]}/${parts[2]}`;

    const email = req.body.email.toLowerCase();
    const about = req.body.about;
    const password = req.body.pass1.toLowerCase();

    db.createUser(name, surname, gender, birthday, email, about,
        password, code)

```

```
        .then(() => res.redirect('/login-en?fromreg=YES'))
        .catch(error => res.status(500).send(error));
    });
```

4. Com es pot veure agafem tots els paràmetres del formulari, es fan les transformacions pertinents i finalment, executem la funció `db.createUser`, la qual es troba en el fitxer `neo4j-api.js`:

```
createUser(name,surname,gender,birthday,email,about,password,code){
    var isadmin = "NO";

    const session = this.driver.session();

    const hash = crypto.createHash('sha256').update(password).digest('base64');

    const resp = session
        .run('
        CREATE (n:USER {
            name: {name},
            surname: {surname},
            gender: {gender},
            birthday: {birthday},
            email: {email},
            about: {about},
            code: {code},
            password: {hash},
            uuid: {uuid},
            isAdmin: {isadmin}
        })
        RETURN n.name', {
            name,
            surname,
            gender,
            birthday,
            email,
            about,
            code,
            hash,
            uuid: uuid(),
            isadmin
        });

    resp.then(() => session.close())
        .catch(() => session.close());

    return resp;
}
```

Aquesta funció encripta la contrasenya en un hash en base64, i seguidament, executa una *query* a la base de dades (vindria a ser `session.run()`). Una vegada s'ha executat la *query*, es retorna el resultat, i com s'ha vist al mètode post del fitxer `app.js`, en cas que la *query* hagi fallat, et retorna l'error 500. En el cas que l'usuari s'hagi creat correctament, et redireccionarà a la pantalla de login.

4.3.3. New Story

Aquesta funcionalitat, dona l'oportunitat de poder penjar una història a l'usuari en qüestió. Per poder implementar-ho, s'ha seguit els següents passos:

1. En primer lloc, s'ha creat un formulari el qual l'usuari ha d'omplir:

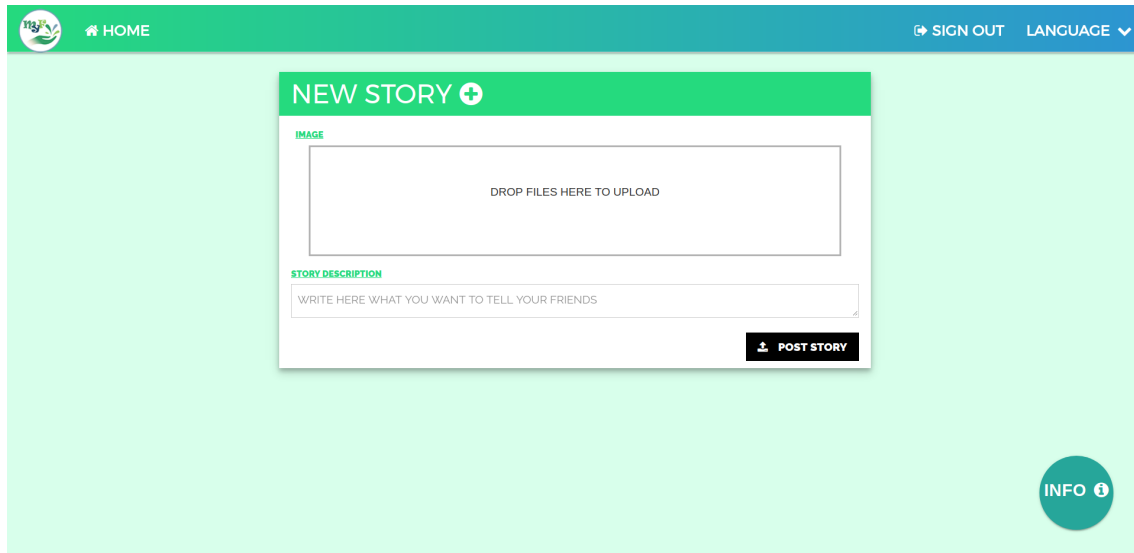


Figura 4.9: Pàgina de New Story.

Com podem veure, podem penjar una imatge i podem ficar una descripció. Per poder penjar la imatge (fent ús del drag & drop), s'ha procedit de la següent forma:

- (a) S'ha definit un formulari amb la classe dropzone i amb identificador #imageform en el fitxer /views/new-post/new-post-en:

```

form#imageform.dropzone.w3-container(action="/post-image", method= 1
    "post", enctype="multipart/form-data")
    label.w3-text(style="margin-top:-45px;margin-left:-40px ! 2
        important;")
        b(style='text-decoration:underline;') IMAGE 3

```

- (b) Una vegada s'ha definit el formulari, es defineix el dropzone en el mateix fitxer:

```

Dropzone.options.imageform = { 1
  2
  // Prevents Dropzone from uploading dropped files 3
  immediately
  autoProcessQueue: false, 4
  maxFiles: 1, 5
  addRemoveLinks: true, 6
  init: function () { 7
    var submitButton = document.querySelector("# 8
      submitbutton")
    myDropzone = this; // closure 9
  }
  submitButton.addEventListener("click", function () { 10
    var form = $("#registerform"); 11
    if(form.valid()) { 12
      $('#spinnerbackground').css('display', 'block' 13
      ); 14
      $('#spinner').css('display', 'block'); 15
      myDropzone.processQueue(); 16
    }
  }
}

```

```

    if (myDropzone.getUploadingFiles().length === 17
        0 && myDropzone.getQueuedFiles().length
        === 0) {
        var description = $("#description").val(); 18
        console.log(description); 19
        $.ajax({ 20
            method: "POST", 21
            url: "/post-story/" + description, 22
            success: function () { 23
                var form = $("#registerform"); 24
                form.submit(); 25
            } 26
        }); 27
    } 28
} 29
}); 30
31

myDropzone.on("complete", function(file) { 32
    if (this.getUploadingFiles().length === 0 && this. 33
        getQueuedFiles().length === 0) {
        var description = $("#description").val(); 34
        console.log(description); 35
        $.ajax({ 36
            method: "POST", 37
            url: "/post-story/" + description, 38
            success: function () { 39
                var form = $("#registerform"); 40
                form.submit(); 41
            } 42
        }); 43
    } 44
}); 45
} 46
}); 47

```

En aquesta definició, es pot veure que una vegada es cliqui el submit button, es pujarà la imatge al servidor (només en el cas que aquesta existeixi).

(c) Quan la imatge es puja al servidor, s'utilitza la següent ruta:

```

app.post('/post-image', uploadPost.single('file'), (req, res) => { 1
    res.send('OK'); 2
}); 3

```

La qual executa la funció `uploadPost.single('file')` definida de la següent forma:

```

const storagePost = multer.diskStorage({ 1
    destination: './uploadsPost/', 2
    filename(req, file, cb) { 3
        crypto.pseudoRandomBytes(16, (err, raw) => { 4
            if (err) return cb(err); 5
            lastUpload = './uploadsPost/${raw.toString('hex')}${path. 6
                extname(file.originalname)}';
            fileUploaded = true; 7
            cb(null, raw.toString('hex') + path.extname(file. 8
                originalname));
        }); 9
    } 10
}); 11
12

var maxSize = 1 * 1000 * 1000*1000*1000; 13
14

const uploadPost = multer({ 15
    storage: storagePost, 16
    limits: { fileSize: maxSize }, 17

```

```
});
```

18

Una vegada s'ha pujat la imatge al servidor, es segueix amb l'execució del codi per publicar la història.

2. Com hem dit anteriorment, una vegada la imatge s'ha pujat al servidor (en el cas que aquesta existeixi), es fa una nova crida al servidor:

```
app.post('/post-story/:description', requireLogin, (req, res) => {
  const description = req.params.description;
  const date = new Date();
  const dateInMilliseconds = date.getTime();
  const usuari = `${req.session.usr[0].name} ${req.session.usr[0].
    surname}`;
  console.log("ehhehehehhehehe");
  if(fileUploaded===true){
    console.log("1");
    db.newPost(req.session.usr[0].email, description, lastUpload
      , dateInMilliseconds, usuari).then(() => {
      console.log("POST");
      db.relationToNewPost(req.session.usr[0].email,
        dateInMilliseconds);
    });
    console.log("LOG");
  }
  else{
    console.log("2");
    db.newPost(req.session.usr[0].email, description, 'nothing',
      dateInMilliseconds, usuari).then(() => {
      console.log("POST");
      db.relationToNewPost(req.session.usr[0].email,
        dateInMilliseconds);
    });
    console.log("LOG");
  }
  fileUploaded=false;
  res.send('OK');
});
```

En aquesta crida es distingeix si anteriorment s'ha pujat una imatge al servidor. Si és així, aquesta es passa com a paràmetre a la funció newPost de la capa de dades. Altrament, aquest paràmetre es passa com a 'nothing'. Finalment, es fa una nova crida a la capa de dades (relationToNewPost), la qual permet crear una relació MY_POST, entre l'usuari i la nova història. Les dues operacions són les següents:

- **newPost():**

```
newPost(email,description,foto, date, usuari){
  const session = this.driver.session();

  const resp = session
    .run('CREATE (n:POST {
      description: {description},
      foto: {foto},
      date: {date},
      usuari: {usuari},
      uuid: {uuid}
    }) RETURN n.uuid', {
      description,
      foto,
      date,
      usuari,
      uuid: uuid(),
    })
```

```
    });  
  
    resp.then(() => session.close())  
        .catch(() => session.close());  
  
    return resp;  
}
```

17
18
19
20
21
22
23

- **relationToNewPost():**

```
relationToNewPost(email,date){  
    const session = this.driver.session();  
  
    var time = new Date().getTime();  
    var query = "MATCH (u:USER), (r:POST) WHERE u.email = '"+  
        email+"' AND r.date = '"+date+"' CREATE (u)-[t:MY_POST {  
            date:'"+time+"' }]->(r) RETURN t";  
  
    const resp = session.run(query);  
  
    resp.then(() => session.close())  
        .catch(() => session.close());  
  
    return resp;  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13

3. Per acabar, es fa una última crida al servidor, la qual redireccionarà al client a la pàgina d'inici.

```
app.post('/redirect-after-post', requireLogin, (req, res) => {  
    res.redirect('/main-page-en');  
});
```

1
2
3

4.3.4. Other User Profile

Aquesta funcionalitat permet poder veure el perfil d'un altre usuari. A diferència de les funcionalitats anteriors, aquesta es correspon a un get del fitxer app.js.

Per fer-ho s'ha procedit de la següent forma:

1. En primer lloc, l'usuari fa la següent crida get al servidor:

```

app.get('/profile-out-en/:targetEmail', requireLogin, (req, res) => { 1
  const email = req.params.targetEmail;                               2
  if (email == req.session.usr[0].email) res.redirect('/profile-en'); 3
  else {                                                                4
    db.findUser(email).then((result) => {                               5
      const usuari = result[0];                                         6
      db.getFriends(email).then((result2) => {                          7
        const friends = result2;                                        8
        db.getMyStories(email, req.session.usr[0].email).then((result3) 9
          => {                                                           10
            const posts = result3;                                      11
            db.getIfFriend(email, req.session.usr[0].email).then((result4 12
              ) => {                                                     13
                var isFriend = false;                                   14
                if(result4[0]!==null) isFriend=true;                   15
                res.render('./profile-out/profile-out-en.pug', { usuari, 16
                  friends, posts, isFriend});                           17
              });                                                       18
            });                                                         19
          });                                                           20
        });
      });
    });
  }
});

```

En primer lloc, comprovem que el perfil que volem veure no sigui el nostre. En cas que el perfil sigui el nostre, s'enrutarà a l'usuari al `app.get("profile-en")`, que es correspon a la ruta del nostre perfil. Seguidament fem les següents crides a la capa de dades:

- `findUser()`:

```

findUser(email){
  const session = this.driver.session();

  const promise = new Promise((resolve, reject) => {
    session
      .run('MATCH (s:USER) WHERE s.email = "' + email + '"
          RETURN s')
      .then((result) => {
        session.close();
        resolve(result.records
          .map(record => record._fields[0].properties)
        );
      })
      .catch((error) => {
        session.close();
        reject(error);
      });
  });

  return promise;
}

```

- `getFriends()`:

```

getFriends(myemail){
  const session = this.driver.session();

  var query = "MATCH (n:USER {email: '"+myemail+"'})-[:
    MY_FRIEND]->(p:USER) RETURN p";

  const promise = new Promise((resolve, reject) => {
    session.run(query)
      .then((result) => {
        session.close();
        resolve(result.records
          .map(record => record._fields[0].
            properties));
      })
      .catch((error) => {
        session.close();
        reject(error);
      });
  });

  return promise;
}

```

- `getMyStories()`:

```

getMyStories(email, myemail){
  const session = this.driver.session();

  var query = 'MATCH (n:USER {email: "' + email + '"})-[:MY_POST
    ]->(p:POST), (r:USER {email: "' + myemail + '"})
    SET p.likes = SIZE()-[:LIKE]->(p)
    SET p.liked = SIZE((r)-[:LIKE]->(p))
    RETURN p';

  const promise = new Promise((resolve, reject) => {
    session
      .run(query)
      .then((result) => {
        session.close();
        resolve(result.records
          .map(record => record._fields[0].properties)
        );
      })
      .catch((error) => {
        session.close();
        reject(error);
      });
  });

  return promise;
}

```

- `getIfFriend()`:

```

getIfFriend(target, myemail){
  const session = this.driver.session();

  var query = "MATCH (n:USER {email: '"+myemail+"'})-[r:
    MY_FRIEND]->(p:USER {email: '"+target+"'}) RETURN r";

  const promise = new Promise((resolve, reject) => {
    session.run(query)
      .then((result) => {
        session.close();
      });
  });
}

```



```
        resolve(result.records
            .map(record => record._fields[0].properties)
        );
    })
    .catch((error) => {
        session.close();
        reject(error);
    });
});

return promise;
}
```

2. Una vegada que s'ha agafat la informació necessària de la capa de dades, es renderitza la pàgina web que el client vol veure passant-li tota aquesta informació.

4.4 Proves

Les proves que s'han dut a terme per validar el *software*, són UT (Unit Test). Els Unit Test, són un tipus de proves que consisteixen en realitzar proves als components més petits de la nostra aplicació, ho sigui, es testeja cada una de les funcionalitats del sistema de forma individual.

Addicionalment, per garantir més rigidesa de l'aplicació, el testeig ha estat realitzat, en part, per companys que també es travaven fent el TFG.

CAPÍTOL 5

Implantació del Sistema

5.1 Introducció

En aquest apartat, es descriurà com implantar el software en cas que alguna associació el vulgui utilitzar amb els seus membres.

S'explicarà com s'implantaria en un servidor Linux, quines lleis s'han de tenir en compte a l'hora d'utilitzar-lo, i com es faria perquè els usuaris el sabessin utilitzar.

5.1.1. Implantació en un servidor Linux

Per poder implantar el sistema en un servidor Linux s'hauran de seguir els següents passos:

1. Descarregar Neo4j del següent enllaç: <https://neo4j.com/download>
2. Extraieu el contingut del fitxer utilitzant la següent comanda:

```
$tar -xf "filecode"
```

1

Per exemple:

```
$tar -xf neo4j-enterprise-2.3.1-unix.tar.gz
```

1

3. Dirigeix-te al fitxer que has extret i executa la següent comanda:

```
$ /bin/neo4j start
```

1

Amb aquesta comanda s'inicialitzarà el servidor de Neo4j. Per defecte s'inicialitzarà a la següent ruta: <http://localhost:7474>

4. Clona el repositori on es troba el codi:

```
$ git clone https://github.com/daleu/makeyourfriend.git
```

1

5. Instal·la Node.js:

```
$ sudo apt-get update
$ sudo apt-get install npm
$ sudo apt-get install nodejs-legacy
```

1

2

3

6. Navega a la carpeta on has descarregat el repositori i executa la següent comanda:

```
$ npm install
```

1

Amb això, s'instal·laran totes les dependències del projecte.

7. Finalment, des de la mateixa ruta, executa la següent comanda:

```
$ node --use_strict app.js
```

1

8. Navega a la següent URL per utilitzar el sistema: <http://localhost:3000>

Addicionalment, s'ha de crear el primer node d'administrador. Per fer-ho, s'ha de fer de la següent forma:

1. Entrar a la ruta <http://localhost:7474>. Aquí es troba la consola de neo4j.
2. En aquesta pàgina informem el camp Username i Password amb "neo4j". Seguidament donem clic a connect. En el cas que demani un canvi de la contrasenya, la canviem, però s'haurà d'anar al fitxer .env del projecte, i informar el camp GRAPHENEDB_BOLT_PASSWORD amb la nova contrasenya.
3. Una vegada estem connectats, executem la següent comanda:

```
CREATE (u:USER {birthday: '06/12/1995', password: '7nmXbJOA1eM3/BwJXs6MjyL5HzBs7rFh+1H+zt4sS6E=', gender: 'Male', isadmin: 'YES', surname: 'Aleu', about: 'I am the admin!', name: 'David', uuid: '3818206-5651-416d-ac34-e49228d6ee7f', email: 'david.aleu@gmail.com'}) RETURN u
```

1

On el camp password, és la contrassenya 11111111 en base 64 utilitzant l'algorisme sha256, i el codi de validació es correspon al camp uuid.

5.1.2. Lleis que apliquen

Pel que fa a les lleis que apliquen en el projecte, les associacions que vulguin utilitzar la plataforma, hauran de tenir en compte la Llei orgànica de protecció de dades de caràcter personal que es poden trobar en aquest URL:

http://administraciojusticia.gencat.cat/web/.content/documents/arxius/loi15_1999lopdc.pdf

Aquestes lleis s'informaran amb anterioritat a les associacions que vulguin fer ús del *software*.

5.1.3. Procediment d'aprenentatge

Una vegada el gestor del sistema ha instal·lat l'aplicació i ha fet els passos necessaris per poder regular totes les lleis, haurà d'ensenyar als usuaris com utilitzar el *software*.

Per fer-ho es començarà amb una sessió, que doni a entendre el propòsit del sistema i com s'ha d'utilitzar. Els responsables de la sessió, estaran ajudant als usuaris en tot moment perquè puguin entendre el comportament del *software*. A més, també s'explicarà els diferents conceptes de la xarxa social, com ara que és una història.

CAPÍTOL 6

Conclusions

Tot aquest treball sorgeix de la idea inicial de crear un *software* que pogués ajudar a la societat. Inicialment, no sabia sobre quin àmbit ho podia fer, però parlant amb una familiar que es dedicava a tractar persones amb diversitat funcional, vaig acabar cedint per ajudar aquest sector, on no existeix gairebé cap *software* que serveixi per ajudar-lo.

Pel que fa al desenvolupament de projecte, s'ha aconseguit implementar totes les funcionalitats generals del projecte. També cal dir que algunes de les funcionalitats definides en el *backlog* inicial no s'han pogut dur a terme per falta de temps, i que a més s'han solapat dues tasques definides en la planificació inicial (Millors amb Documentació i Defensa), tot i que s'ha pogut recuperar el ritme establert.

Pel que fa al resultat final del projecte, crec que és molt millorable, ja que personalment, la vista gràfica de la pàgina web no és molt agradable. Tot hi això, s'ha aconseguit l'objectiu principal del projecte, que és que la interfície gràfica sigui usable per persones amb diversitat funcional de dèficit cognitiu.

Com a conclusions personals, donat que mai m'havia enfrontat a un projecte de tal envergadura tot sol, han sigut moltes les vegades que he pensat que no podria acabar el projecte a causa d'un problema que no sabia resoldre. Tot hi això, a mesura que anava avançant el treball, veia que els problemes s'anaven resolent a poc a poc, i finalment he pogut acabar amb tot el que s'havia establert inicialment.

El treball de final de grau m'ha ajudat a aprendre i adquirir nous coneixements. No era la primera vegada que m'enfrontava a tecnologies web, ja que anteriorment havia programat amb Ruby on Rails. Però el *framework* utilitzat en aquest projecte (Node.js) era completament nou per a mi, i per tant, puc dir que m'ha donat a adquirir molts coneixements nous, els quals segurament pugui utilitzar en el meu futur laboral.

6.1 Futurs Projectes

Com s'ha dit anteriorment, el projecte és molt millorable, per tant, he pensat una sèrie de propostes per a que aquest es pugui seguir. Les propostes són les següents:

- **Modificació de la interfície gràfica:** Una de les coses principals que no m'agrada del resultat final del projecte és la seva interfície gràfica. Ara mateix, si fes el projecte de nou, segurament el que faria primer seria definir les classes necessàries perquè la interfície quedés en un estat agradable a la vista de l'usuari. Per tant, un dels futurs projectes que he pensat, és la modificació de la interfície gràfica i l'addició d'alguna funcionalitat (com ara poder comentar les diferents històries de la xarxa social).
- **Creació d'una aplicació per a mòbil:** Per tal que l'aplicació desenvolupada sigui més usable, he pensat en que un dels futurs projectes sigui el desenvolupament de l'aplicació per a mòbil. La tecnologia mòbil cada vegada avança més ràpid, i per tant, tenir una aplicació mòbil pot servir en un futur pròxim.
- **Connexió amb l'API d'una xarxa social:** Com s'ha dit anteriorment, un dels objectius d'aquest projecte és que els seus usuaris no es sentin un públic diferent. Per tant, un dels futurs projectes que es pot dur a partir d'aquest, és crear una connexió amb l'API d'alguna xarxa social, mantenint tot el que està fet actualment. Per exemple, les històries que es publiquen a l'aplicació, es publiquin com a posts a facebook, i els likes d'aquests també es vegin plasmades en l'aplicació. D'aquesta forma s'aconseguiria una doble funcionalitat. Una capa gràfica adaptada per utilitzar facebook, i una eina de gestió amb informació d'aquest.

Bibliografia

- [1] Discapacitat cognitiva. Consultat a <https://neetic.jimdo.com/part-te%C3%B2rica/discapacitats/discapacitat-cognitiva/>.
- [2] Xarxa social - Viquipèdia, l'enciclopèdia lliure. Consultat a https://ca.wikipedia.org/wiki/Xarxa_social.
- [3] Historia de las redes sociales - Recursos - Ministerio de Educación. Consultat a <http://recursostic.educacion.es/observatorio/web/es/internet/web-20/1043-redes-sociales?start=2>.
- [4] Bulletin Board System - Viquipèdia, l'enciclopèdia lliure. Consultat a https://ca.wikipedia.org/wiki/Bulletin_Board_System.
- [5] Aplicaciones para niños con Necesidades Educativas Especiales. Consultat a <http://www.chaval.es/chavales/educacion/aplicaciones-para-ni%C3%B1os-con-necesidades-educativas-especiales>.
- [6] Stimulus – APP profesional de estimulación cognitiva. Consultat a <http://stimuluspro.com/>.
- [7] 10 apps para alumnos con dificultades específicas de aprendizaje. Consultat a <http://www.educaciontrespuntocero.com/accesibilidad/apps-alumnos-dificultades-especificas-aprendizaje-dea/25080.html>.
- [8] Twitter - Viquipèdia, l'enciclopèdia lliure. Consultat a <https://ca.wikipedia.org/wiki/Twitter>.
- [9] Facebook - Viquipèdia, l'enciclopèdia lliure. Consultat a <https://ca.wikipedia.org/wiki/Facebook>.
- [10] Instagram - Viquipèdia, l'enciclopèdia lliure. Consultat a <https://ca.wikipedia.org/wiki/Instagram>.
- [11] LinkedIn - Viquipèdia, l'enciclopèdia lliure. Consultat a <https://ca.wikipedia.org/wiki/LinkedIn>.
- [12] Chart: Facebook Users Show No Signs of Tiring | Statista. Consultat a <https://www.statista.com/chart/5380/facebook-user-engagement/>.
- [13] Unitat de Tècniques Augmentatives de Comunicació (UTAC). Consultat a <http://www.utac.cat/>.
- [14] Associació Lectura Fàcil. Consultat a <http://www.lecturafacil.net/>.
- [15] ARASAAC: Portal Aragonés de la Comunicación Aumentativa y Alternativa. Consultat a <http://www.arasaac.org/>.

-
- [16] HTML - Wikipedia, la enciclopedia libre. Consultat a <https://es.wikipedia.org/wiki/HTML>.
 - [17] HTML Tutorial. Consultat a <https://www.w3schools.com/html/>.
 - [18] CSS Tutorial. Consultat a <https://www.w3schools.com/css/>.
 - [19] JSON - Viquipèdia, l'enciclopèdia lliure. Consultat a <https://ca.wikipedia.org/wiki/JSON>.
 - [20] Neo4j, the world's leading graph database - Neo4j Graph Database. Consultat a <https://neo4j.com/>.
 - [21] WebStorm: The Smartest JavaScript IDE. Consultat a <https://www.jetbrains.com/webstorm/>.
 - [22] Node.js. Consultat a <https://nodejs.org/en/>.
 - [23] expressjs/body-parser: Node.js body parsing middleware. Consultat a <https://github.com/expressjs/body-parser>.
 - [24] Bootstrap · The world's most popular mobile-first and responsive front-end framework. Consultat a <http://getbootstrap.com/>.
 - [25] dotenv. Consultat a <https://www.npmjs.com/package/dotenv>.
 - [26] jQuery. Consultat a <https://jquery.com/>.
 - [27] jQuery Validation Plugin | Form validation with jQuery. Consultat a <https://jqueryvalidation.org/>.
 - [28] Getting Started – Pug. Consultat a <https://pugjs.org/api/getting-started.html>.
 - [29] Font Awesome, the iconic font and CSS toolkit. Consultat a <http://fontawesome.io>.
 - [30] W3.CSS Home. Consultat a <https://www.w3schools.com/w3css>.
 - [31] crypto. Consultat a <https://www.npmjs.com/package/crypto>.
 - [32] client-session. Consultat a <https://www.npmjs.com/package/client-session>.
 - [33] expressjs/multer: Node.js middleware for handling 'multipart/form-data'. Consultat a <https://github.com/expressjs/multer>.
 - [34] Moment.js | Docs. Consultat a <https://momentjs.com/docs/>.
 - [35] FullCalendar - JavaScript Event Calendar. Consultat a <https://fullcalendar.io/>.
 - [36] Documentation - Materialize. Consultat a <http://materializecss.com/>.